

UNIVERSITAT DE LLEIDA

ESCOLA POLITÈCNICA SUPERIOR

Grau en Enginyeria Informàtica



Universitat de Lleida
Escola Politècnica Superior

Treball Final de Grau

Liquid Galaxy POIs Controller

Presentat per:

Rafael Durán Sàez

Tutora:

Teresa Alsinet Bernadó

Lleida, 2015

Índex

Índex de Figures	3
Índex de taules	4
Agraïments	5
Resum.....	6
Introducció	7
Google Summer of Code 2015	8
Estructura del GSoC.....	8
Liquid Galaxy	9
TFG – GSoC 2015	10
Propostes.....	11
Liquid Galaxy POIs Controller	11
Wikipedia Liquid Galaxy Mashup	13
Aprenentatge previ	14
Elecció dels participants GSOC 2015	14
Liquid Galaxy POIs Controller	15
Objectius	15
Temporització (<i>timeline</i>)	16
Anàlisis previ	17
Requeriments.....	18
Riscs	19
Cas d'ús.....	20
Tecnologies.....	21
Base de Dades	21
Estructura, disseny i funcionament de l'aplicació.....	28
Usuari bàsic	28
Usuari administrador.....	33
Creació.....	35
Modificació i Eliminació	40
Problemes i Solucions	44
Extensions al TFG en el marc de GSoC	48
Avaluació, valoració i conclusions	49
Bibliografia	50

Índex de Figures

Figura 1: Google Summer of Code 2015	8
Figura 2: Mostra d'ús del Liquid Galaxy	9
Figura 3: Exemples Liquid Galaxy	9
Figura 4: Primer disseny interfície POIs i Tours.....	11
Figura 5: Disseny interfície extra.....	12
Figura 6: Disseny Interfície Wikipedia Mashup.....	13
Figura 7: Cas d'ús Liquid Galaxy POIs Controller.....	20
Figura 8: Diagrama de classes UML.....	22
Figura 9: Entitat POI del Diagrama UML	23
Figura 10: Entitat Tour del Diagrama UML.....	24
Figura 11: Entitat Category del Diagrama UML.....	25
Figura 12: Entitat Tour POIs del Diagrama UML	26
Figura 13: Interfície inicial Usuari Bàsic.....	28
Figura 14: Icona de cerca manual	28
Figura 15: Vista de POIs.....	29
Figura 16: Vista de tots els Tours	30
Figura 17: Vista de Tours per categories.....	30
Figura 18: Ruta de categories i tipus de vista.....	31
Figura 19: Elements per categoria	31
Figura 20: Back i Back to Start.....	32
Figura 21: Administration Tools contrassenya.....	32
Figura 22: Interfície Usuari Administrador.....	33
Figura 23: Editar, Eliminar o Cancel·lar	33
Figura 24: Botó per crear elements.....	35
Figura 25: Text per crear POI, Tour o Categoria.....	35
Figura 26: Nou element dins de la categoria actual.....	35
Figura 27: Nou POI	36
Figura 28: Nova Categoria	37
Figura 29: Nou Tour.....	38
Figura 30: Afegir POI al Tour	38
Figura 31: Llista de POIs en un Tour.....	39
Figura 32: Eliminar POI del Tour.....	39
Figura 33: Modificar o Eliminar POI	40
Figura 34: Modificar POI.....	40
Figura 35: Modificar Tour.....	41
Figura 36: Modificar Categoria.....	42
Figura 37: Exemple codi ListViews	45
Figura 38: Exemple codi modificar subcategories.....	47

Índex de taules

Taula 1: Exemple Categories	26
Taula 2: Exemple Tour POIs - Taula de Tours	27
Taula 3: Exemple Tour POIs - Taula de POIs	27
Taula 4: Exemple Tour POIs - Taula de Tour POIs	27
Taula 5: Exemple Modificar Categoria - Taula Category	42
Taula 6: Exemple Modificar Categoria - Taula POIs	43
Taula 7: Exemple Modificar Categoria - Taula Category Actualitzada	43
Taula 8: Exemple Modificar Categoria - Taula POIs Actualitzada	43

Agraïments

Primer de tot vull donar les gràcies a la meva tutora Teresa Alsinet per oferir-me l'opció de realitzar el Treball Final de Grau vinculat amb el Google Summer of Code 2015; a l'Andreu Ibañez per confiar en mi, incloure'm en el Lleida Liquid Galaxy LAB i permetre'm participar en el GSoC 2015; als meus companys i amics de classe per debatre amb mi les opcions i dificultats trobades durant la fase consistent en l'elecció de TFG; als meus amics per donar-me ànims per tirar endavant el projecte, tant amb l'enfocament del GSoC com el del TFG; i per últim i probablement el més profund, a la meva família i la meva parella per motivar-me, animar-me i aportar-me confiança absoluta per emprendre i endinsar-me en aquest projecte.

Resum

Aquest informe consisteix en la memòria del projecte [Liquid Galaxy POIs Controller](#) [1], realitzat com a **Treball Final de Grau** a partir de la participació en el programa [Google Summer of Code 2015](#) [2]. Es basa en una aplicació fluida pel sistema operatiu Android que permetrà a l'usuari seleccionar els punts d'interès (POIs) i els Tours (rutes d'un conjunt de POIs) per poder visualitzar-los en les pantalles d'un sistema [Liquid Galaxy](#) [3]. A més a més, l'aplicació també permetrà a l'administrador d'aquesta crear, modificar i eliminar POIs i Tours.

La memòria ha estat redactada sota les normes d'estil IEEE [4] per a facilitar la comprensió de la seva lectura.

Introducció

Per tal de contextualitzar la proposta del TFG, al llarg d'aquesta memòria, presentarem el programa Google Summer of Code (durant la memòria també referenciat com a “GSoC”). Així doncs, introduïrem els conceptes de GSoC, algunes dades històricament significatives, l'estructura que el compona, la seva organització, el procediment per a escollir els membres que en formaran part i la meua relació amb el programa. També explicarem breument què és el projecte Open Source Liquid Galaxy (d'aquí en endavant anomenat també com “LG”), en què consisteix, la relació amb els dos projectes que vam presentar al GSoC 2015 (Liquid Galaxy POIs Controller, el qual va ser finalment escollit, i Wikipedia Liquid Galaxy Mashup) i algunes de les seves característiques. Posteriorment, essent el nucli d'informació més rellevant d'aquesta memòria, expliquem les tasques realitzades prèviament a l'inici del projecte com són l'aprenentatge de tecnologies i llenguatges nous, la presa de requeriments i riscos, l'establiment dels objectius del projecte i el càlcul de la línia de temps que guia aquest. Definim l'estructura de la base de dades de l'aplicació i exposem exemples del seu funcionament, expliquem el disseny i l'estructura de l'aplicació, tant en aspectes a nivell d'usuari com a nivell d'implementació, detallem l'evolució de l'aplicació juntament amb els problemes sorgits i les corresponents solucions i finalment determinem les tasques que s'han de realitzar per finalitzar la implementació de l'aplicació juntament amb una avaluació del treball realitzat.

Tenim el deure de comentar que les llistes definitives dels participants en el GSoC no es van publicar fins a finals d'abril, per tant al no saber fins la data exacta si estàvem admesos o no, ens hem vist obligats a esperar fins llavors per a poder començar el TFG. Juntament amb la meua tutora, **Teresa Alsinet**, hem acordat que entregariem i exposariem el contingut que tinguéssim realitzat fins les dates corresponents al dipòsit de la memòria ja que aquest programa és una opció que ofereix la Universitat de Lleida perquè els alumnes realitzin el seus TFG i, d'una forma o altra, s'han de compensar totes les dates. Per tant, recalquem que el Treball Final de Grau no avarca tot el contingut del projecte presentat al GSoC 2015, sinó el contingut desenvolupat fins la data del dipòsit de la memòria. Tot i així, aquest es continuarà desenvolupant fins finalitzar-lo dins dels límits establerts pel propi GSoC 2015.

Google Summer of Code 2015

És un programa a nivell mundial en el qual **Google** ofereix a estudiants universitaris la possibilitat, en cas de ser escollits, d'obtenir una beca (5.500\$) pel desenvolupament de software vinculat amb diversos projectes Open Source sota la supervisió d'un mentor prèviament assignat.



Figura 1: Google Summer of Code 2015

Aquest programa està en actiu des de 2005 i ha comptat amb més de 7.500 estudiants de més de 100 països arreu del món. En l'edició d'aquest any han estat escollits aproximadament 1.000 estudiants dels 7.000 presentats, 3 dels quals pertanyents a la Universitat de Lleida (Marc Solé, Marc González i jo) i un altre a la Universitat Rovira i Virgili de Tarragona (Julio Bondia).

Estructura del GSoC

Per a poder participar en el GSoC s'han de realitzar una sèrie de tasques per anar passant els diferents talls i aconseguir finalment la beca:

1. Entrar en contacte amb una de les empreses col·laboradores amb el projecte, en el nostre cas **Ponent 2002**.
2. Escollir d'entre els possibles projectes Open Source per desenvolupar el teu sub-projecte. En el nostre cas el **Liquid Galaxy (LG)**.
3. Presentar la/les teves propostes de projecte i un *timeline*. En el nostre cas **Liquid Galaxy POIs Controller** i **Wikipedia Liquid Galaxy Mashup**.
4. En cas de ser escollit (inici de Maig) començar a programar fins a final d'estiu.

Liquid Galaxy

És un dels projectes Open Source escollit per GSoC perquè els estudiants participants en el programa puguin desenvolupar sub-projectes d'aquest. Es basa en un clúster d'ordinadors en els quals s'executa el Google Earth de forma sincronitzada i permet a l'usuari veure el contingut en un conjunt de pantalles en un format *tot-en-un* creant una experiència d'immersió increïble.

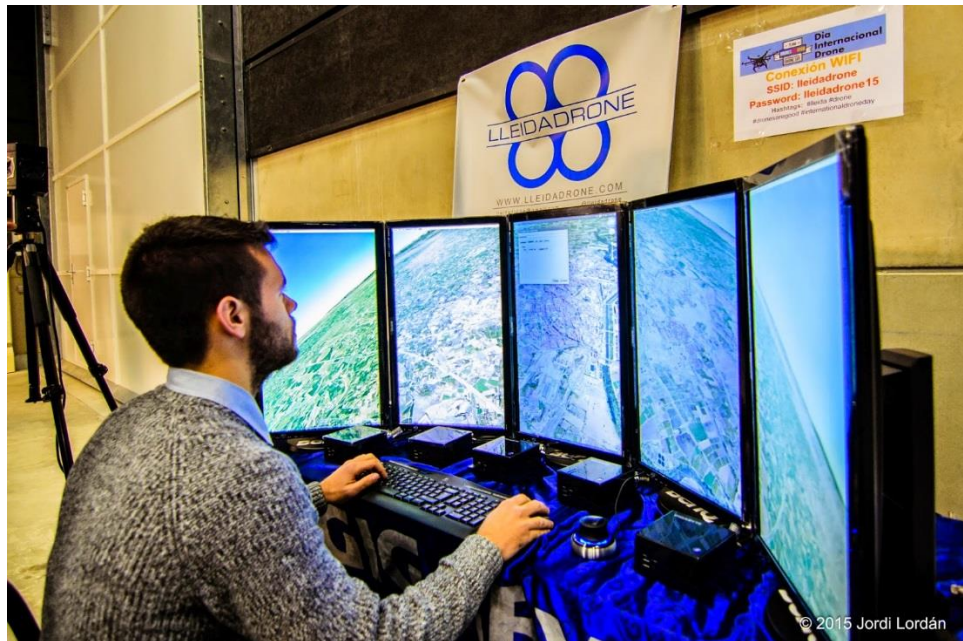


Figura 2: Mostra d'ús del Liquid Galaxy



Figura 3: Exemples Liquid Galaxy

TFG – GSoC 2015

En el nostre cas, durant el mes de gener, la Universitat de Lleida a través de la professora **Teresa Alsinet** va oferir la possibilitat de realitzar el TFG participant en aquest programa. Vam interpretar-ho com una gran oportunitat per aprendre i millorar en molts aspectes de diferents camps de la informàtica i del món laboral i vam decidir intentar-ho. Per tant, vam anar al laboratori **Lleida Liquid Galaxy LAB** [5] per tractar el tema amb l'**Andreu Ibáñez** (coordinador del programa a Lleida).

A partir d'aquí, un cop definida la participació en el GSoC 2015, va començar el nostre Treball Final de Grau. Els passos inicials serien els següents:

1. **Propostes diferents pel Liquid Galaxy.**
2. **Aprenentatge de tecnologies i llenguatges nous.**
3. **Elecció dels participants definitius en el GSOC 2015.**
4. **Començar amb la implementació del projecte.**

Gràcies al treball realitzat en els mesos anteriors a l'elecció dels membres participants en el GSoC 2015, el qual es troba especificat en un apartat [posterior](#), finalment vaig ser escollit com un dels estudiants participants i d'aquesta forma presentaria el Treball Final de Grau com el projecte a desenvolupar en el Google Summer of Code 2015.

Propostes

Per a tenir més opcions de poder entrar en el GSoC 2015, vam decidir, amb l'Andreu Ibáñez, presentar 2 projectes diferents: el **Liquid Galaxy POIs Controller** i el **Wikipedia Liquid Galaxy Mashup**.

Liquid Galaxy POIs Controller

Consisteix en desenvolupar un controlador de punts d'interès (POIs) del LG en forma d'aplicació fluïda per Android. Aquesta interfície d'usuari té tres seccions principals:

- **Selector de POIs i Tours**

Permet a l'usuari seleccionar els punts d'interès o *tours* que vol veure en les pantalles del LG i enviar-los al sistema mitjançant un túnel SSH.

- **Cerca per teclat i veu**

Introduint, tant per veu o pel teclat del dispositiu Android, el destí que es vol veure en els monitors del LG.

- **Eines de gestió**

Aquesta opció permet als usuaris administradors crear, editar i eliminar els POIs i Tours. També facilita a l'administrador realitzar diverses funcionalitats bàsiques per gestionar el LG com són tancar la plataforma, ressetejar el LG, llençar altres aplicacions, etc. Per accedir a totes aquestes eines de gestió és necessària la introducció d'una contrasenya.

El primer 'disseny' de la interfície va ser:

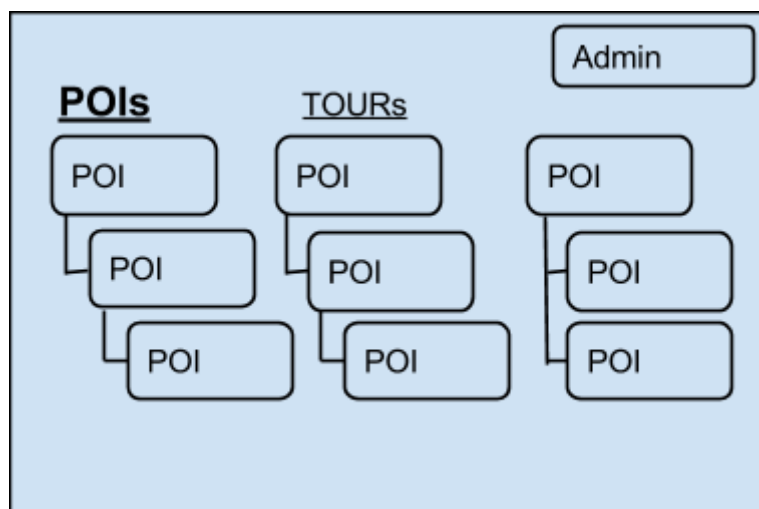


Figura 4: Primer disseny interfície POIs i Tours

En cas que el desenvolupament de l'aplicació evolucioni ràpid, com a funcionalitat extra dividiríem la interfície principal de l'aplicació en dues parts: una per la selecció de POIs i Tours tal com es mostra en la figura anterior i l'altra mitja part l'ocuparia un mapa (de Google Maps) per poder seleccionar en el propi mapa la localització que es desitja veure en els monitors del LG. El 'disseny' seria aquest:

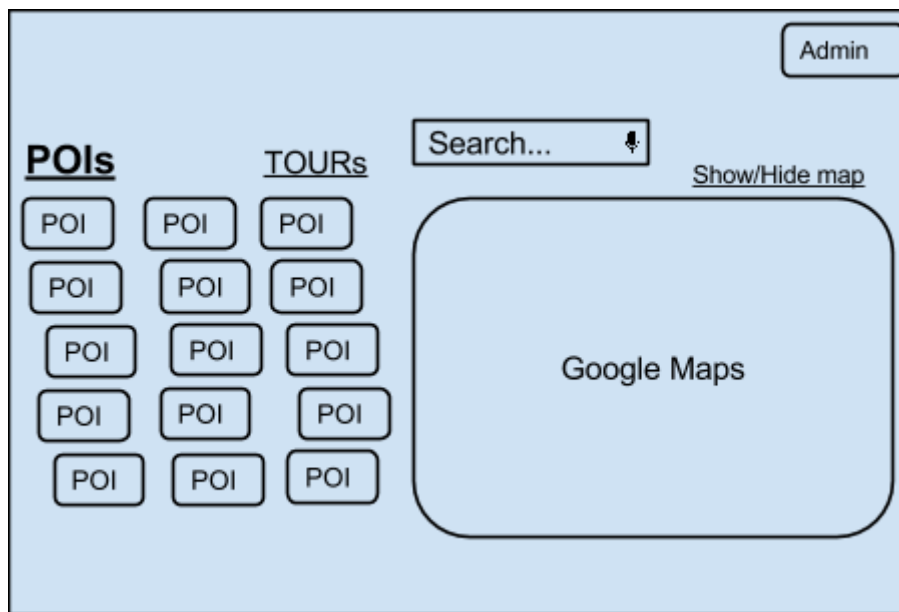


Figura 5: Disseny interfície extra

El seus valors per la comunitat són:

- Actualment existeix una interfície web per seleccionar POIs però no per editar-los ni organitzar Tours, per tant el seu funcionament no és del tot correcte. Realitzar aquesta aplicació comporta satisfer una necessitat per al món del Liquid Galaxy ja que facilitaria molt l'ús del software. D'aquesta forma substituiríem la interfície actual i milloraríem les funcionalitats d'ús i gestió del LG.
- Ampli ús en tabletas Android, amb millor resposta al toc de les pantalles tàctils.
- Es consideraria un valor afegit la creació d'un manual en línia explicant el procés de captura de POIs.

Wikipedia Liquid Galaxy Mashup

Es basa en crear un vincle entre l'API de Wikimedia i el Liquid Galaxy. L'enllaç permet mostrar en els monitors del LG, entre d'altres, la ubicació de l'últim article carregat en Wikipedia i una bombolla amb la seva informació, un Tour de les últimes imatges publicades, etc.

Per aconseguir aquest objectiu s'ha de desenvolupar una aplicació fluïda per Android que s'encarrega de connectar-se amb l'API, obtenir i transformar les dades desitjades i finalment enviar-les al LG a través d'un túnel SSH. La idea és consultar l'API amb un interval de temps establert per l'usuari i mostrar la informació en els monitors del LG. En la interfície de l'aplicació, l'usuari té l'opció de filtrar les dades per: país, idioma i cronològicament.

El primer 'disseny' de la interfície del prototip de l'aplicació era aquest:

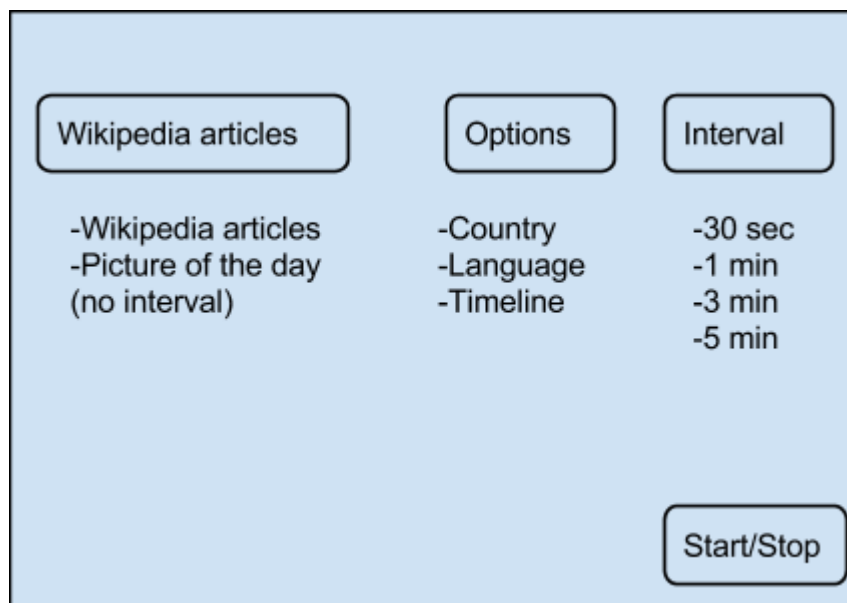


Figura 6: Disseny Interfície Wikipedia Mashup

El seus valors per la comunitat són:

- És una gran eina informativa pels visitants.
- Ús lúdic per visitants o administradors.
- Ús com a salva pantalla quan els LG estan en desús.

Aprenentatge previ

Tot allò relacionat amb el Liquid Galaxy ens resultava totalment desconegut fins a l'inici del TFG, per tant vam haver d'investigar i aprendre diversos coneixements per poder realitzar correctament el projecte:

- **Funcionament del Liquid Galaxy:** què és, com funciona internament, tipus de connexions amb l'exterior, capes que s'hi poden aplicar, etc.
- **Llenguatge Android:** personalment, mai havia programat amb Android, per tant per a poder desenvolupar l'aplicació vam haver d'aprendre'n mitjançant dos cursos *on-line* que ensenyaven les característiques bàsiques de la programació amb Android i, en el nostre cas, aprofundiment en certes temàtiques. Els cursos són:
 - **Udacity – Developing Android Apps** [6] (realitzat per Google).
 - **Miriada X – Android, Programación de Aplicaciones** [7].

Durant la fase d'aprenentatge, es va implementar una aplicació anomenada Sunshine [8].

- **Accés a les dades de Wikipedia:** per poder obtenir les dades que proporciona Wikipedia havíem d'estudiar com obtenir-les de l'API de Wikimedia i com tractar-les per a la seva posterior mostra a través dels monitors del LG.

Tot aquest procés ens va mantenir ocupats durant uns 3 mesos, fins a finals d'abril que va ser quan es van publicar les llistes dels estudiants admesos al programa.

Elecció dels participants GSOC 2015

El 27 d'abril van sortir les llistes dels estudiants que formarien part definitivament del GSOC 2015. El meu nom constava en la llista i aquell fet va marcar un punt i seguit per continuar endavant tant amb el TFG com amb el GSOC 2015.

El nostre projecte escollit va ser el del **Liquid Galaxy POIs Controller**. Sincerament, preferíem aquest abans que l'altre ja que aquesta aplicació és una eina necessària pel món del Liquid Galaxy que facilitarà moltíssim el seu ús i gestió, en canvi el projecte vinculat amb la Wikipedia podríem descriure'l més aviat com una aplicació lúdica.

Liquid Galaxy POIs Controller

El desenvolupament d'aquest projecte no solament es basa en la implementació del codi de l'aplicació sinó que està format per un conjunt de tasques que ajuden a que el desenvolupament del projecte sigui eficient. Aquestes estan explicades en les següents seccions: [Objectius](#), [Temporització](#), [Anàlisi](#), [Tecnologies](#), [Base de Dades](#), [Estructura](#), [Problemes](#) i [Tasques futures](#).

Objectius

Els objectius principals del projecte **Liquid Galaxy POIs Controller**, descrits breument a la secció [anterior](#), són els següents:

- Desenvolupar una aplicació fluida per Android que funcioni com a controlador dels punts d'interès (POIs) i Tours que es desitgen mostrar i veure en els monitors del Liquid Galaxy.
- L'usuari haurà de ser capaç de seleccionar el POI o Tour que desitgi amb un sol clic i veure'l en el LG.
- L'usuari haurà de ser capaç d'introduir, tant per veu com per teclat, el destí que desitja veure en el LG.
- L'administrador haurà de ser capaç de crear, editar i eliminar POIs, Tours i Categories (en l'apartat [Estructura de l'aplicació](#) s'explica més extensament).
- L'administrador haurà de ser capaç d'executar funcionalitats bàsiques del Liquid Galaxy des de la pròpia aplicació: ressetejar el LG, tancar-lo, aplicar-hi diferents capes, etc.
- S'hauran de poder agregar POIs mitjançant la importació d'arxius en format *csv* que continguin les dades necessàries per la creació d'aquests.

A més a més, l'execució d'aquest projecte també té com a objectiu incorporar una sèrie de beneficis i interessos personals com són l'aprenentatge d'un nou llenguatge: Android; millora en la cerca d'informació, en el desenvolupament de projectes, en la implementació de software i un notable descobriment i aprenentatge de noves formes de treballar.

Temporització (*timeline*)

A principis de febrer, respectant les dates clau establertes per la direcció del GSoC 2015, vam distribuir temporalment les tasques a fer pel desenvolupament del projecte. A continuació podem observar la línia de temps creada i els punts principals a complir en cada un dels períodes establerts:

- **Abans del 27 d'Abril**
Aprendre sobre les funcionalitats del Liquid Galaxy i programació en Android.
- **27 Abril – 25 Maig (Període crucial)**
Discutir els detalls del projecte amb el mentor.
Decidir les tecnologies que s'han d'utilitzar.
Determinar l'estructura del projecte i reunions amb el mentor i la tutora.
Desenvolupar un prototip.
- **25 Maig – 3 Juliol (Primer període de desenvolupament – Avaluació a mig termini)**
Implementar tot allò relacionat amb l'estructura de l'aplicació (activitats, fragments, *layouts*...), implementar la base de dades i la majoria de funcionalitats de l'aplicació.
Redactar memòria del Treball Final de Grau.
- **3 Juliol – 17 Agost (Segon període de desenvolupament)**
Finalitzar la memòria del TFG i dipositar-la
Implementar les últimes funcionalitats i enllaçar-ho amb el Liquid Galaxy.
Dissenyar la interfície d'usuari.
Si hi ha temps, implementar la funcionalitat extra amb l'API de Google Maps.
- **17 Agost – 21 Agost**
Finalitzar la documentació per al GSoC 2015.

Com indica la línia de temps anterior, la feina corresponent a la implementació de l'aplicació comença a finals de Maig (tot i que nosaltres ja vam començar a principis de mes realitzant un prototip) perquè així ho estableix GSoC i no és fins llavors que es pot iniciar el desenvolupament. Per aquest motiu i tal com es descriu en la [introducció](#) de la memòria, el TFG solament abasteix les funcionalitats de l'aplicació desenvolupades fins la data de dipòsit i aquesta no romandrà finalitzada fins finals d'estiu.

Anàlisi previ

Tal com durant el curs 2013/2014 vam estudiar en l'assignatura "Enginyeria de Requeriments", el conjunt d'un projecte es basa en 3 fases a realitzar en el següent ordre: l'Anàlisi, el Desenvolupament i el Manteniment i Millora. L'anàlisi d'un projecte consisteix en la presa dels requeriments i riscos, tant tecnològics i socials com funcionals i no funcionals, que comporta el desenvolupament d'un projecte. També implica l'elaboració d'un model gràfic (en el nostre cas del tipus Model UML) de les entitats o elements més rellevants del projecte a desenvolupar, juntament amb una esquematització d'algunes de les situacions o accions que el producte final del projecte presenta (casos d'ús). Aquesta fase s'ha de realitzar prèviament al desenvolupament i amb molta cura ja que una correcta planificació evita molts errors i permet la possibilitat de tornar enrere durant diverses fases de l'execució del projecte. El desenvolupament de l'aplicació consisteix en la implementació d'aquesta respectant les restriccions, normes i funcionalitats establertes en la fase d'anàlisi. Per últim, la fase de millora i manteniment es basa en aconseguir que el producte del projecte (en el nostre cas l'aplicació) es mantingui actualitzat en referència a les necessitats de l'usuari i als objectius establerts de forma que apareguin millores puntuals i s'eviti l'aparició de qualsevol tipus de error o defecte en el producte. També es basa en la confecció de la part vinculada amb l'evolució del producte.

A continuació exposem tot el contingut relacionat amb l'Anàlisi del projecte, identificant els [requeriments](#) genèrics per a la realització del projecte, separant els tecnològics dels socials per així tenir clar quins aspectes centrar únicament en el desenvolupament del projecte i quins altres abasten matisos relacionats amb el dia a dia. A més a més, també identifiquem els requeriments funcionals, aquells que corresponen a les funcionalitats que el sistema ha de permetre, i els no funcionals, relacionats amb qualitats generals que ha de posseir l'aplicació. Expliquem els [riscos](#) trobats durant l'anàlisi i definim el [cas d'ús](#) que esquematitza l'aplicació. Pel referent al diagrama de classes UML hem de mencionar que, tot i que la seva localització correcta seria dins d'aquest apartat, estarà situat en l'apartat de [Base de dades](#) per així poder relacionar millor tots els elements i aconseguir que els exemples siguin més clars.

Requeriments

El conjunt de tots aquests classificats pels àmbits esmentats són:

Requeriments Genèrics Tecnològics

- Ús de l'entorn Android Studio (Eclipse també serveix però té menys funcionalitats).
- Coneixement del llenguatge Android i la connexió SSH.
- Disposar d'un *tablet* de 7" o 10" amb sistema operatiu Android versió mínima 4.0 (mínim sdk 14) per a realitzar les comprovacions i tests necessaris mentre es desenvolupa l'aplicació.
- Disposició d'un sistema Liquid Galaxy per a la connexió amb l'aplicació.
- Tenir instal·lat Google Earth.
- GitHub per al control de versions de l'aplicació

Requeriments Genèrics Socials

- Temps mínim de 8 hores diàries de dedicació al desenvolupament de l'aplicació i memòria del TFG.
- Reunions amb la tutora del TFG en les dates establertes.
- Reunions setmanals amb el mentor del GSoC.
- Col·laboració amb el Lleida Liquid Galaxy LAB.

Requeriments Funcionals

- L'aplicació ha de permetre a l'usuari bàsic clicar en l'aplicació i veure en els monitors del LG qualsevol POI o Tour.
- L'aplicació ha de permetre veure els POIs classificats per categories
- L'aplicació ha de permetre veure tots els Tours o veure'ls classificats per categories.
- L'aplicació ha de permetre a l'administrador veure totes les Categories o veure-les classificades per altres categories.
- L'aplicació ha de permetre a l'administrador crear POIs, Tours i Categories.
- L'aplicació ha de permetre a l'administrador editar qualsevol característica dels POIs, Tours i Categories.
- L'aplicació ha de permetre a l'administrador eliminar POIs, Tours i Categories.

- L'aplicació ha de permetre a l'administrador executar funcionalitats del LG
- L'aplicació ha de permetre a l'usuari bàsic cercar POIs escrivint per teclat.

Requeriments No Funcionals

Dins dels requeriments no funcionals considerarem els de rendiment, usabilitat, seguretat, legals, eficiència i fiabilitat.

- **Rendiment:**
El sistema ha de tenir una capacitat mínima d'emmagatzemament de 500 POIs, Tours i Categories.
- **Usabilitat:**
La interacció usuari-aplicació ha de ser senzilla i ha de facilitar l'aprenentatge de l'usuari. La UI ha de ser agradable i respectar les directrius de Material Design.
- **Seguretat:**
Per accedir a les funcionalitats de l'administrador, el sistema haurà d'obligar a l'usuari a introduir una contrasenya correcta. A més a més, la contrasenya ha d'estar encriptada.
- **Legal:**
El sistema ha de respectar les directrius Open Source.
- **Eficiència:**
El temps de resposta ha de ser breu i adequat per a una continua i correcta interacció usuari-aplicació.
- **Fiabilitat:**
L'aplicació ha de ser consistent amb un bon control de gestió i errors.

Riscs

El fet d'acceptar aquest projecte (i repte a la vegada) personalment implicava deixar la feina com a programador a l'empresa Indra Sistemas ja que tan l'aprenentatge previ com sobretot el desenvolupament del projecte requerien molt temps i a la vegada produirien una fatiga considerable. També existia el risc que la direcció del GSoC 2015 no acceptés cap dels nostres projectes presentats i per tant haguéssim perdut el temps d'espera a la publicació de quins projectes serien escollits, de la mateixa forma que no haguéssim pogut gaudir d'un mentor (per part del GSoC) per a la resolució dels dubtes durant la implementació de l'aplicació.

Cas d'ús

L'objectiu d'aquest és detectar les funcionalitats que podrà fer el sistema, identificar els actors que hi intervindran i crear les relacions existents entre aquests juntament amb una organització adequada al funcionament de l'aplicació. Per complir aquest objectiu, hem creat el següent model gràfic que representa el que s'acaba de descriure, on podem observar que un Usuari Bàsic simplement serà capaç de veure els POIs i TOURS existents i de registrar-se com a Usuari Administrador. Aquest, a part de visualitzar els POIs i TOURS també serà capaç de visualitzar les Categories, podrà crear, editar i eliminar tot element pertanyent als tres tipus mencionats i, a més a més, executar un conjunt de funcionalitats bàsiques per a la gestió del Liquid Galaxy.

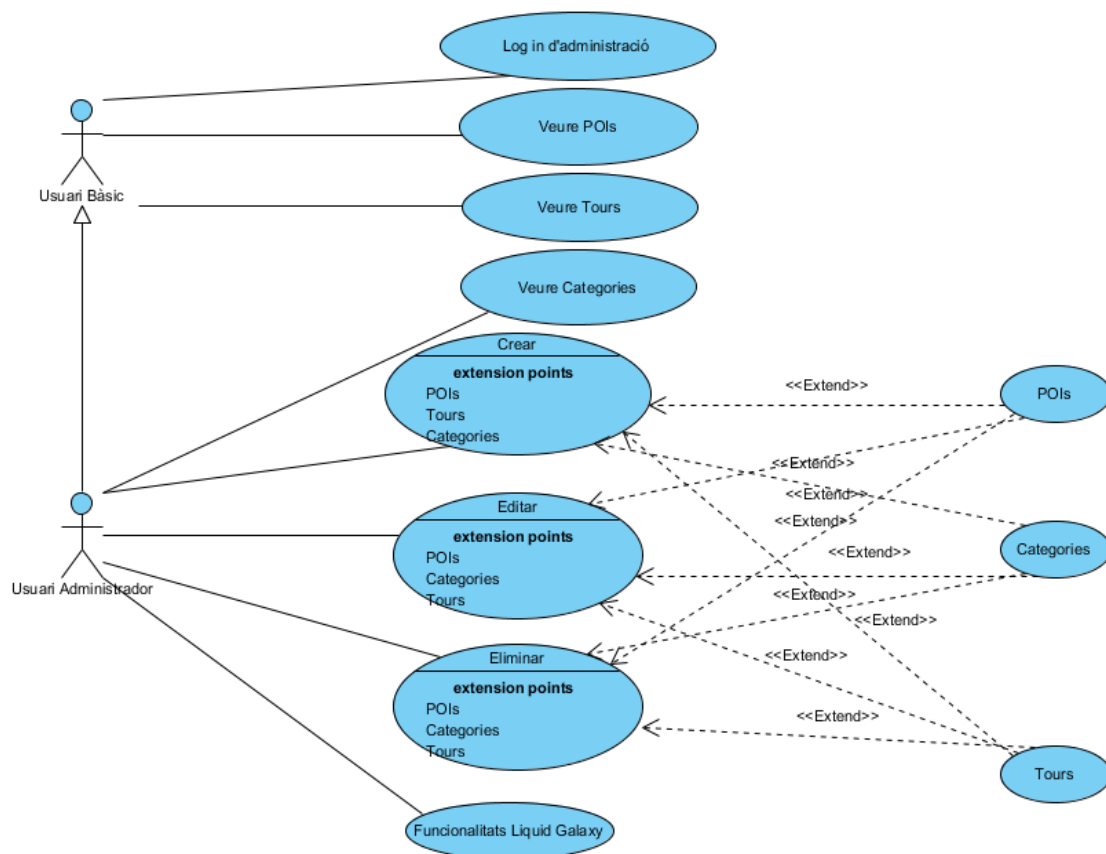


Figura 7: Cas d'ús Liquid Galaxy POIs Controller

Tecnologies

Les tecnologies que inicialment es van pensar utilitzar per al complet desenvolupament de l'aplicació són les següents:

- **Android Studio, SQLite, bash, SSH, Google Earth:** desenvolupament de l'aplicació.
- **UML Visual Paradigm:** diagrama UML i requeriments.

Base de Dades

Per la persistència de la base de dades hem utilitzat **SQLite** ja que va incorporat dins de l'entorn **Android Studio**. Per la implementació i estructuració d'aquesta ens ajudem dels anomenats *Content Providers* pels diversos motius següents:

- Gestionen l'accés a un conjunt estructurat de dades de forma que les encapsulen i proporcionen mecanismes per a la seguretat d'aquestes.
- Són la interfície estàndard que connecta les dades d'un sol procés amb codi que s'executa en un altre procés.
- Presenten les dades a aplicacions externes com una o més taules que són similars a les taules que es troben en una base de dades relacional. Una fila representa una instància d'algun tipus de dades que el proveïdor recopila, i cada columna de la fila representa una peça individual de les dades recollides per una instància.
- Permeten compartir informació de forma segura i eficient més enllà dels límits de l'aplicació abstraient-se de la font de dades subjacent, sigui SQLite (com en el nostre cas), arxius o qualsevol altra cosa amb la finalitat que altres aplicacions puguin accedir a ella sense la necessitat de comprendre com s'han emmagatzemat les dades.
- Tot i que no es vulgui compartir informació amb altres aplicacions, el seu ús és molt útil. Jo utilitzo SQLite, però podria estar emmagatzemant dades en arxius, dinàmicament o en una biblioteca de bases de dades diferent, llavors és molt més fàcil canviar eventualment la font de dades i molt més fàcil per altres administrar la capa de codi de la interfície d'usuari sense tenir que comprendre en profunditat la implementació de l'emmagatzemament de dades.
- Així és com funcionen els *widgets* de Google Play Store i Gmail.

Estructura i Diagrama de classes UML

Per descriure la base de dades que emprà l'aplicació per persistir les dades, vam elaborar un diagrama de classes UML que facilités l'estructuració d'aquesta i permetés veure les relacions entre totes les entitats. El diagrama és el que es pot veure en la figura que ve a continuació:

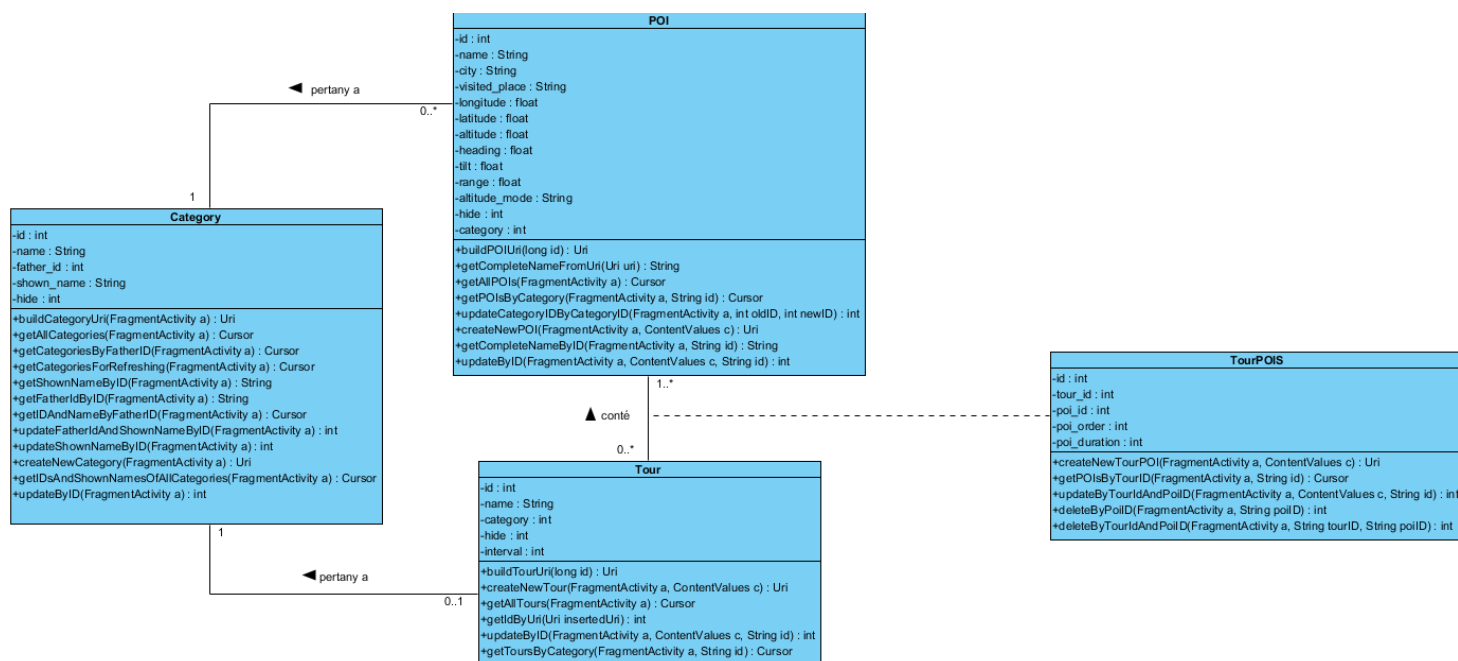


Figura 8: Diagrama de classes UML

El diagrama és simple i es basa en els elements principals de l'aplicació. Cada entitat correspon a una taula amb el mateix nom en la base de dades pel que fa als aspectes d'implementació. Algunes de les operacions de cada entitat les vam plantejar abans de començar la implementació, és a dir, un cop es feia l'anàlisi previ, en canvi n'hi ha que les hem afegit a mesura que anava desenvolupant l'aplicació. Les quatre entitats que componen l'UML s'expliquen més detalladament a continuació:

- **POI**

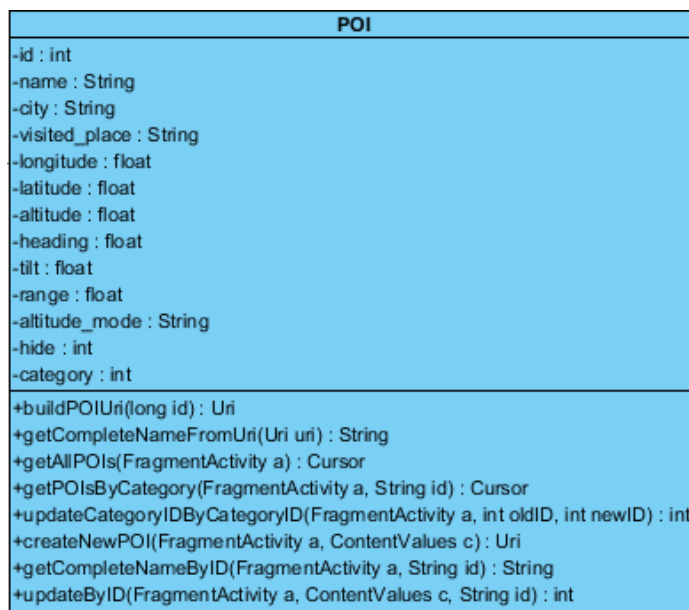


Figura 9: Entitat POI del Diagrama UML

L'entitat de la figura fa referència a tot punt d'interès que es desitgi guardar en el sistema per a la seva posterior visualització en els monitors del LG. Conté els camps que s'indiquen a la figura, els quals signifiquen el següent: *city* i *visited_place* fan referència al nom de la ciutat i el lloc visitat, la unió dels quals compona l'atribut *name* ($name = city + visited_place$); un atribut per cada dada corresponent a la localització del POI: *longitude*, *latitude* i *altitude*; quatre atributs més per les característiques posicionals de la càmera que mostra el punt desitjat: *heading*, *tilt*, *range* i *altitude_mode*; l'atribut *hide* perquè l'administrador decideixi si el POI s'ha de mostrar o no en la llista de punts d'interès que l'usuari pot veure, i un últim atribut *category* fent referència al ID de la categoria a la que pertany. L'identificador de cada registre és un número enter (*int*) que és clau primària i va auto incrementant-se cada cop que s'afegeix una nova fila. Cada registre d'aquesta taula està vinculat, mitjançant l'atribut i clau forana *category*, amb l'atribut *id* de la taula *Category*. L'atribut *category* té un valor per defecte de 0 (per si no hagués cap categoria creada) mentre que l'atribut *name* ha de ser únic. Per a la persistència d'un punt d'interès són necessàries absolutament totes aquestes dades i, per tant, l'absència del valor d'alguna d'aquestes no és possible, és a dir, cap atribut dels que el formen pot tenir, com anomenem en el llenguatge informàtic, valor *null*.

- **Tour**

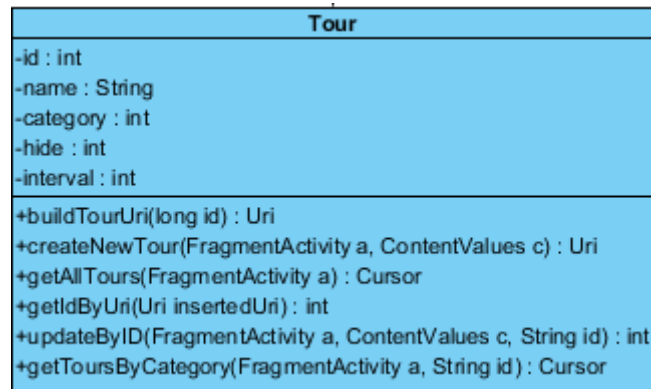


Figura 10: Entitat Tour del Diagrama UML

Referència a tot Tour que es desitgi crear i guardar en el sistema per a la seva posterior visualització en el LG. Els camps indicats en la figura representen el següent: l'atribut *name* fa referència al nom que se li vulgui donar al Tour, els atributs *hide* i *category* tenen la mateixa funció que en els POIs i l'atribut *interval* correspon al temps (en segons) que cada POI es mostrarà en el LG, és a dir, l'interval de temps que l'usuari desitja que existeixi, en general, entre POI i POI d'un Tour. Dic “en general” perquè , tal com explico més tard dins d'aquest apartat en la justificació de la taula “Tour POIs”, l'usuari també pot establir un interval de temps diferent per a cada punt d'interès. Per tant, aquest atribut dóna valor a la duració de tot aquell POI (dins d'un Tour) que l'usuari no hagi indicat un interval en concret.

Tal com succeeix amb els POIs, l'identificador de cada registre d'aquesta taula és un número enter (*int*) que és clau primària i va auto incrementant-se cada cop que s'afegeix una nova fila, cada registre d'aquesta taula està vinculat, mitjançant l'atribut i clau forana *category*, amb l'atribut *id* de la taula *Category* i cap dels atributs pot tenir valor *null*.

- **Category**

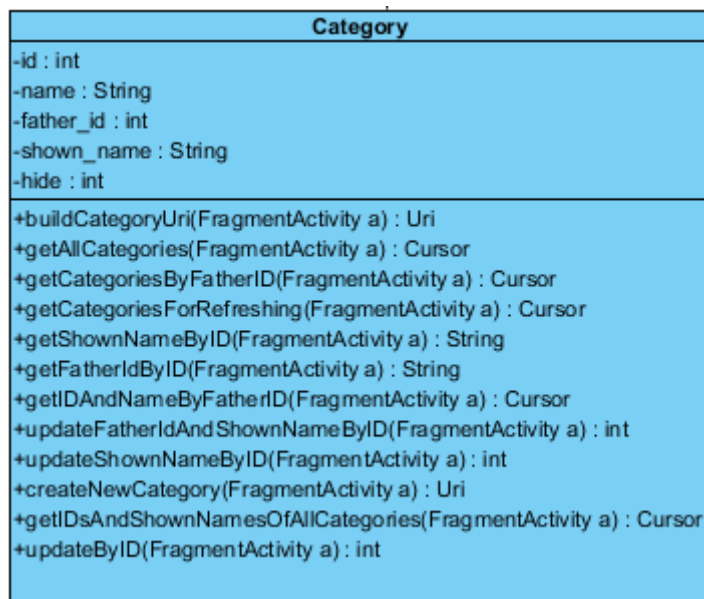


Figura 11: Entitat Category del Diagrama UML

Tot registre d'aquesta taula significa una categoria a la qual poden pertànyer els punts d'interès i tours, d'aquesta forma aquests romandran classificats segons les preferències de l'administrador de l'aplicació. Pel referent als camps de la figura: el nom de la categoria correspon a l'atribut *name*; l'atribut *fatherID* correspon a l'identificador de la categoria pare o superior; *shown_name* fa referència al nom de tota la ruta de categories superiors fins arribar a l'actual i *hide* té la mateixa funció que en els POIs i Tours.

De la mateixa forma que amb els POIs i Tours, l'identificador de cada registre és un número enter (*int*) que és clau primària i va auto incrementant-se cada cop que s'afegeix una nova fila i cap atribut pot tenir un valor null. L'atribut *shown_name* és únic ja que servirà per diferenciar possibles categories amb el mateix nom, és a dir, pot ser que existeixin dues categories amb l'atribut *name* = "Hotels", però una es pot referir als hotels de Lleida i l'altra als hotels de Madrid. Per diferenciar-ho existeix aquest camp que permetrà referenciar-nos a una categoria o una altra.

Un exemple simple per entendre-ho millor:

Name	ID	Father_ID	Shown_name
Catalunya	1	0	Catalunya/
Lleida	2	1	Catalunya/Lleida/
Madrid	3	0	Madrid/
Hotels	4	2	Catalunya/Lleida/Hotels/
Hotels	5	3	Madrid/Hotels/

Taula 1: Exemple Categories

- **Tour POIs**

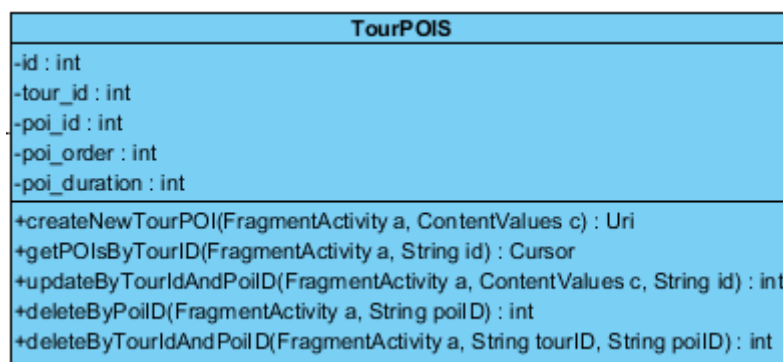


Figura 12: Entitat Tour POIs del Diagrama UML

Aquesta entitat fa referència a la relació entre un POI i el Tour pertanyent, és a dir, es refereix a tot POI contingut en un Tour i serveix per vincular-los amb els Tours on estan inclosos (no tots els POIs tenen l'obligació d'estar inclosos en algun Tour). Els atributs *tour_id* i *poi_id* serveixen per identificar el Tour i el POI respectivament, *poi_order* correspon a l'ordre que té un POI dins del Tour al qual pertany i *poi_duration* fa referència al temps que l'usuari vol que aquell POI en aquell Tour en concret es vegi en el LG. Un POI pot estar vinculat amb 0 o molts Tours. L'identificador de cada registre és un número enter (*int*) que és clau primària i va auto incrementant-se cada cop que s'afegeix una nova fila. Cada registre d'aquesta taula està vinculat, mitjançant l'atribut i clau forana *Tour*, amb l'atribut *id* de la taula *Tour*, i mitjançant l'atribut i clau forana *POI*, amb l'atribut *id* de la taula *POI*. Com passa també amb les altres taules, cap dels atributs pot tenir valor null.

Per a entendre-ho millor, les següents taules representen un simple exemple:

Tour

ID	Name	Interval	Altres camps
1	Lleida Lovely Places	20	...
...

Taula 2: Exemple Tour POIs - Taula de Tours

POIs

ID	Visited place	Altres camps
1	Seu Vella	...
2	Camp d'Esports	...
3	Escola Politècnica Superior	...
4	Parc Científic Gardeny	...
...

Taula 3: Exemple Tour POIs - Taula de POIs

TourPOIs

ID	Tour ID	POI ID	POI Order	POI Duration
1	1	1	2	10
2	1	2	3	20
3	1	3	1	15
4	1	4	4	20
...	

Taula 4: Exemple Tour POIs - Taula de Tour POIs

El Tour amb el nom “Lleida Lovely Places” estarà compost dels punts d’interès següents: “Escola Politècnica Superior”, “Seu Vella” i “Camp d’Esports” ordenats respectivament, tal com es pot observar en les figures anteriors. La duració de cada POI depèn de si l’usuari, en l’aplicació, indica simplement un interval genèric (en aquest cas 20 segons) o també indica un interval per certs POIs (en aquest cas 10 pel POI amb ID = 1 i 15 pel ID = 3).

Estructura, disseny i funcionament de l'aplicació

L'aplicació es divideix en dues parts principals, la de l'usuari bàsic i la de l'administrador.

Usuari bàsic

Aquesta es basa en la selecció de POIs i Tours per part de l'usuari sense drets de gestió ni administració i la mostra d'aquests en els monitors del Liquid Galaxy.

En la següent figura podem observar com en la part superior hi ha dos botons perquè l'usuari pugui clicar i veure tots els POIs i Tours que l'aplicació conté.



Figura 13: Interfície inicial Usuari Bàsic

També podem observar que a l'alçada dels botons esmentats però a la dreta de la figura hi ha una icona en forma de lupa, com la ressaltada en vermell en la figura següent:

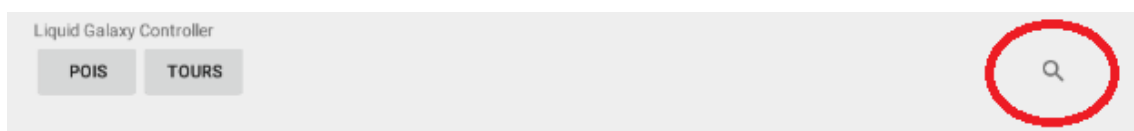


Figura 14: Icona de cerca manual

Aquesta serveix per a que l'usuari introdueixi manualment quin destí vol veure en els monitors del Liquid Galaxy.

Retornant el tema dels botons, si l'usuari clica en el botó amb el text "TOURS" apareixerà un llistat amb tots els continguts en l'aplicació, però si clica en el de "POIS" els veurà classificats per categories. Per tant, quan l'usuari cliqui en el botó corresponent als punts d'interès, veurà aquelles categories que no en tenen cap de superior i únicament aquells POIs que no pertanyen a cap categoria. Un cop l'usuari cliqui en una categoria, veurà els POIs d'aquesta categoria i les subcategories d'aquesta. En canvi, quan l'usuari cliqui en el botó dels Tours els veurà absolutament tots i també podrà decidir veure'ls classificats per categories. Amb els POIs no té sentit que es vegin tots perquè és de suposar que l'aplicació en contindrà una quantitat molt elevada i carregar-los tots no seria gens eficient.

Les vistes de POIs i Tours serien com les de les figures següents:

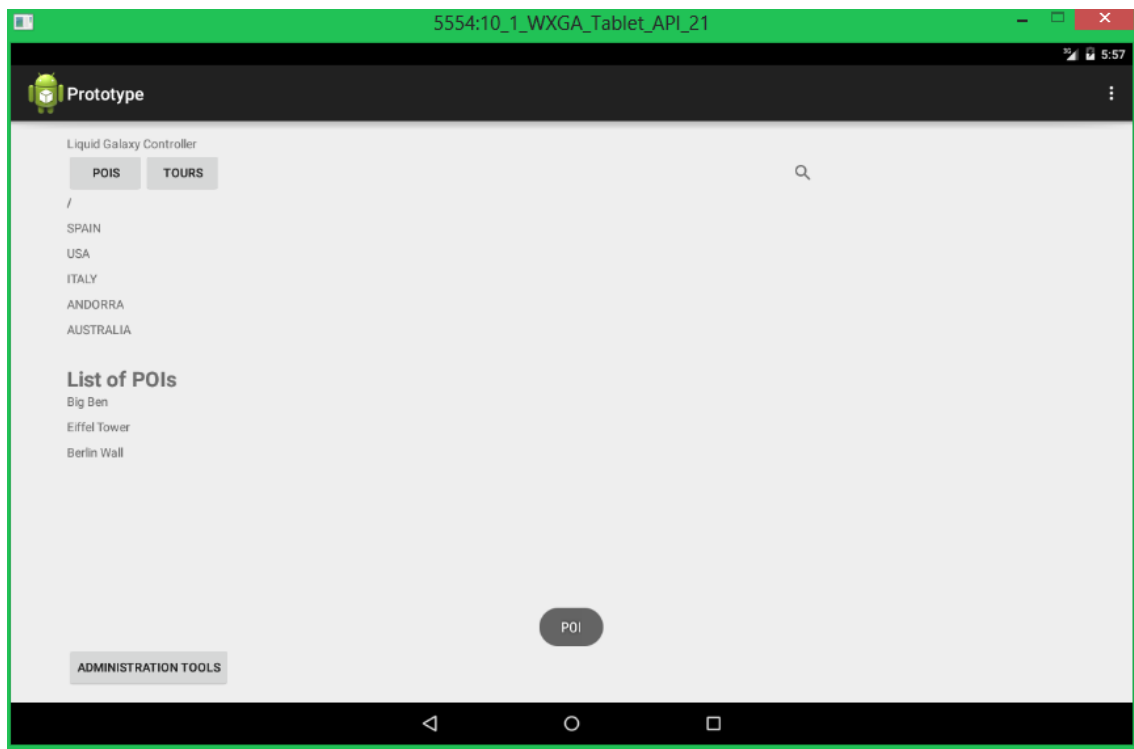


Figura 15: Vista de POIs

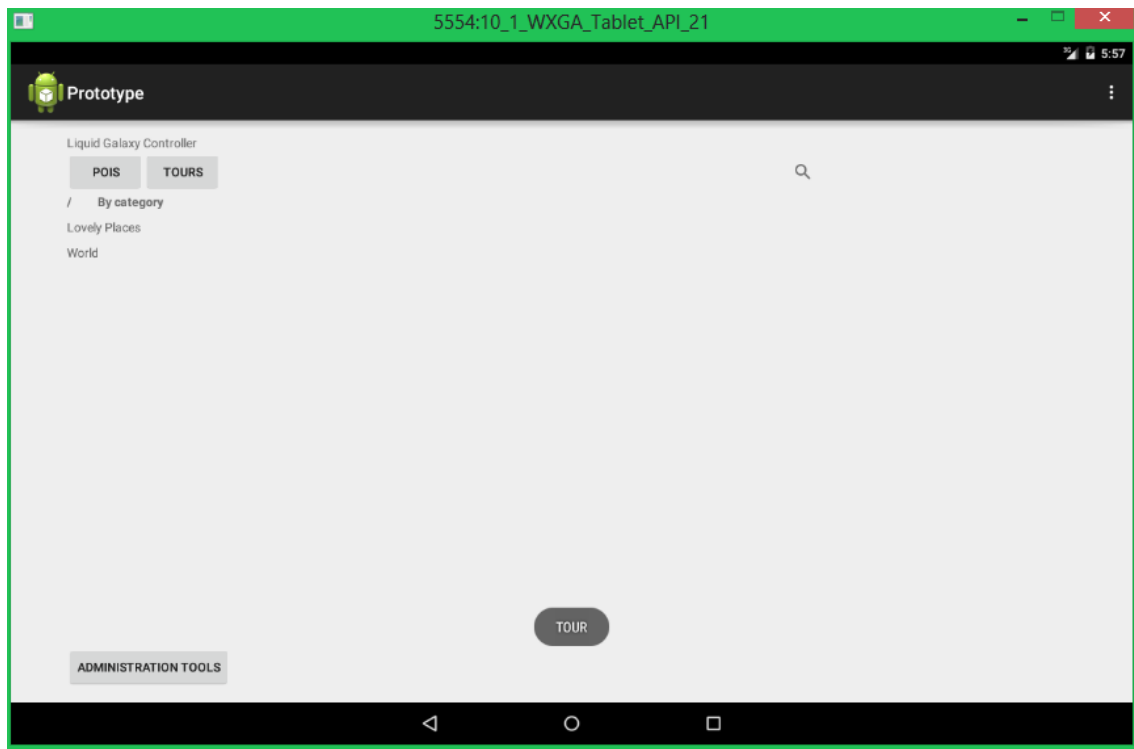


Figura 16: Vista de tots els Tours



Figura 17: Vista de Tours per categories

En les figures anteriors podem observar que sobre la llista de POIs/Tours o sobre la llista de categories quan veiem els elements classificats per aquestes, hi ha dos camps de text similars als de la següent figura:



Figura 18: Ruta de categories i tipus de vista

El primer text (vermell) correspon a la ruta de categories que s'està mostrant. En la darrera figura s'indica que som dins de la categoria "SPAIN" però en les tres figures anteriors veiem que aquests camps simplement contenen el text "/". Això és degut a que, com que era el primer cop que l'usuari clicava en el botó de POIS o TOURS, els elements que es mostraven eren els que no pertanyien a cap categoria, per tant no tenen cap ruta de categories. El text "/" en aquest camp també apareix quan l'usuari decideix veure absolutament tots els Tours, donat que no hi ha cap categoria per la qual filtrar.

El segon (verd) és clicable i solament apareix en el cas dels Tours. Permet a l'usuari veure tots els Tours clicant en *All* o veure'ls classificats per categoria clicant en *By Category*.

En la següent figura podem observar els nous elements que apareixen quan l'usuari clica sobre una categoria:

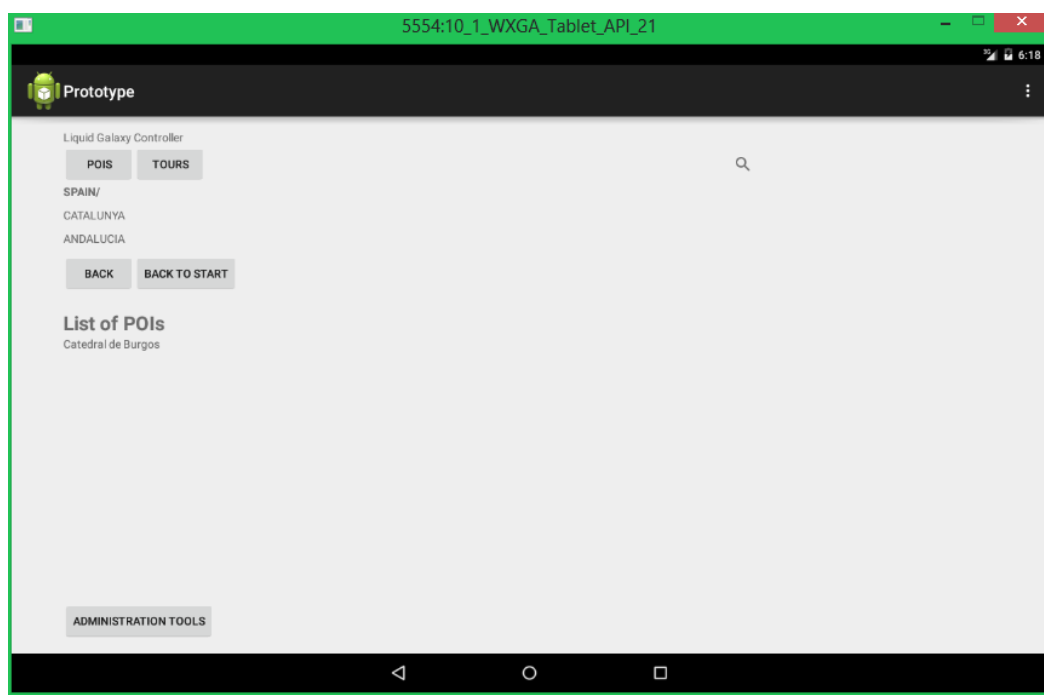


Figura 19: Elements per categoria

Si observem la ruta de categories veiem que som dins de la categoria “SPAIN” i que l’únic POI dins d’aquesta és “Catedral de Burgos”. Podria ser que dins de les subcategories de “SPAIN” hi hagués més POIs però per veure’ls s’ha d’entrar en aquestes.

Quan l’usuari clica en un dels POIs/Tours que apareixen en la llista, aquest es mostra en els monitors del LG, en canvi si prem una de les categories es mostren tots els POIs/Tours inclosos en dita categoria i les subcategories d’aquesta. Si aquesta categoria no té POIs/Tours es mostra una llista buida i un missatge indicant-ho. Si ens fixem, quan som dins d’una categoria, també apareixen dos botons amb el text “Back” i “Back to start”, com en la figura següent:



Figura 20: Back i Back to Start

Aquests serveixen per tornar a la vista corresponent a la categoria anterior o per tornar a l’inici de la vista per categories.

També podem observar que en la cantonada sud-oest hi ha un botó amb el text “ADMINISTRATION TOOLS”. Quan es fa un clic sobre aquest botó apareix una petita finestra en pantalla demanant una contrasenya per accedir a l’apartat de gestió i administració de POIs o, en altres paraules, per accedir a la part de l’Usuari Administrador.

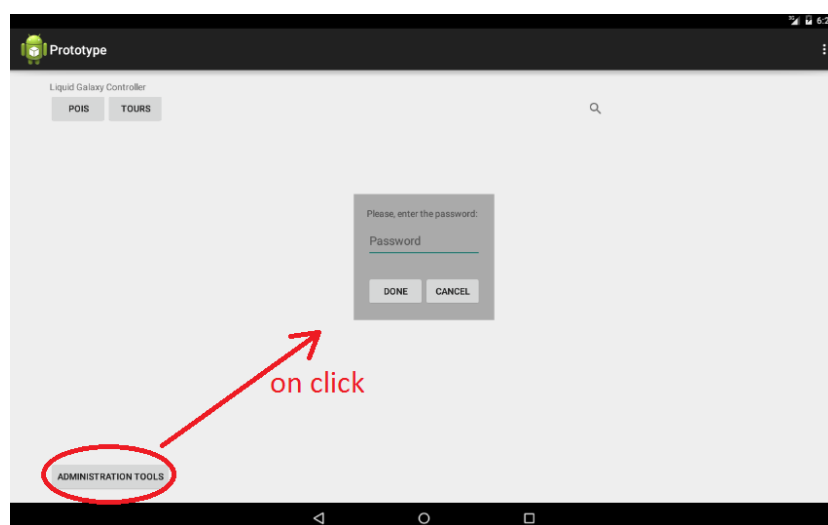


Figura 21: Administration Tools contrassenya

Usuari administrador

Aquesta part es basa en 3 accions bàsiques: **creació**, **modificació** i **eliminació** de POIs, Tours i Categories.

En la figura podem observar que en la pantalla principal, a la qual s'accedeix després d'inserir la contrasenya correcta en la finestra que s'obre al clicar el botó "Administration Tools" de l'apartat Usuari Bàsic, apareixen quatre botons: un per la gestió de **POIs**, un altre per la gestió de

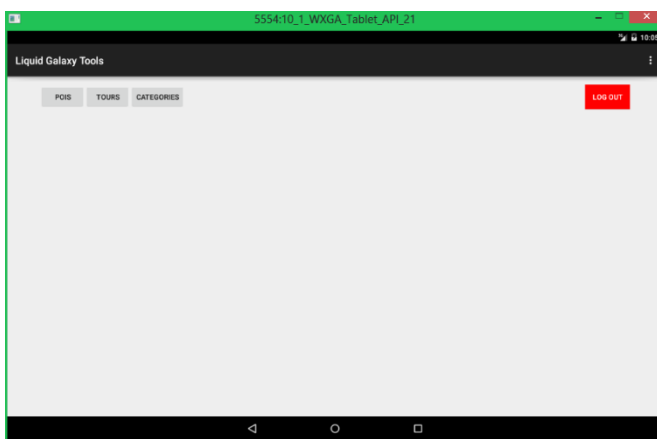


Figura 22: Interfície Usuari Administrador

Tours, un altre per la de **Categories** i un últim (vermell i amb el text "LOG OUT") en la cantonada nord-est per sortir de l'apartat d'Usuari Administrador i tornar a Usuari Bàsic.

Quan es prem el botó de **POIs** o **Tours** es mostren tots aquests de la mateixa forma que se li mostren a l'Usuari Bàsic: un llistat amb tots els elements guardats en l'aplicació amb la possibilitat de veure'ls filtrats per categories i navegar entre elles. Tot i així, la diferència és que al prémer un POI o Tour, enlloc de mostrar-se en els monitors del LG, apareix una petita finestra amb dos botons per **Modificar** o **Eliminar** el ítem seleccionat i un altre botó per cancel·lar l'operació. Aquestes opcions s'expliquen més endavant en l'apartat [Modificació i Eliminació](#).

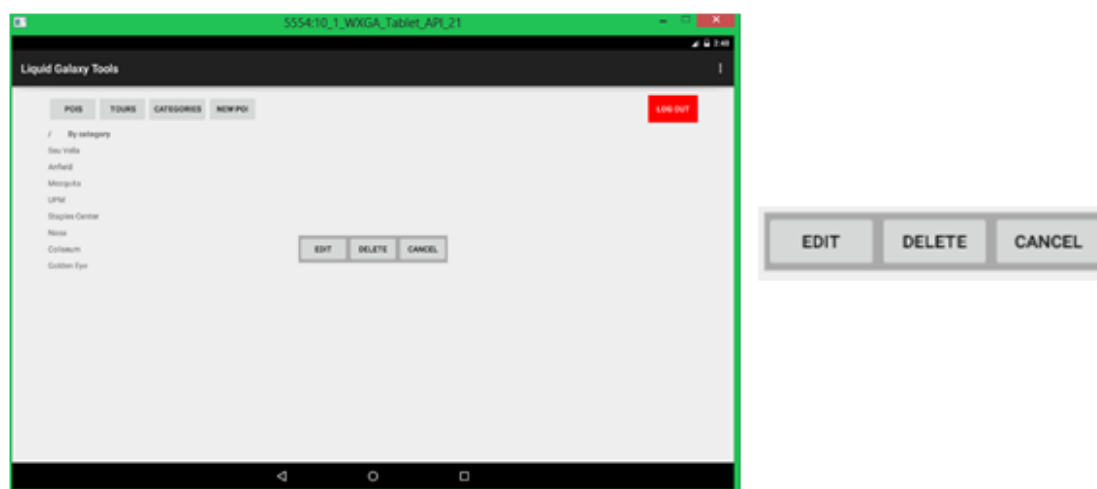


Figura 23: Editar, Eliminar o Cancel·lar

Pel que fa a les categories que es mostren quan es selecciona la vista per categoria (clic en *By Category*), en prémer-n'hi una el comportament de l'aplicació és el mateix que en l'apartat d'Usuari Bàsic.

Un cop l'administrador prem el botó de **Categories** també es mostren totes de forma que es pot escollir, de la mateixa manera que amb els Tours, veure-les totes o filtrades per categoria. Un cop es clica un ítem (una categoria) de la llista de categories poden succeir dues alternatives:

1. Si s'estan mostrant totes les categories existents en l'aplicació, després d'un clic simple, tal com passa amb els POIs i Tours, es mostra una petita finestra amb les opcions de Modificar, Eliminar i cancel·lar l'operació.
2. Si s'estan mostrant les categories filtrades (únicament es veuen les subcategories de la seleccionada), quan fem un clic simple en un ítem de la llista es mostraran les subcategories d'aquesta i així successivament. En canvi, si fem un clic llarg, sortirà la petita finestra (Modificar, Eliminar, Cancel·lar) mencionada en l'alternativa 1.

A continuació explicarem com es realitzen les funcionalitats de [Creació](#) i de [Modificació i Eliminació](#) de POIs, Tours i Categories.

Creació

Quan entrem tant en l'apartat de POIs com el de Tours o Categories, apareix un botó per crear nous ítems:

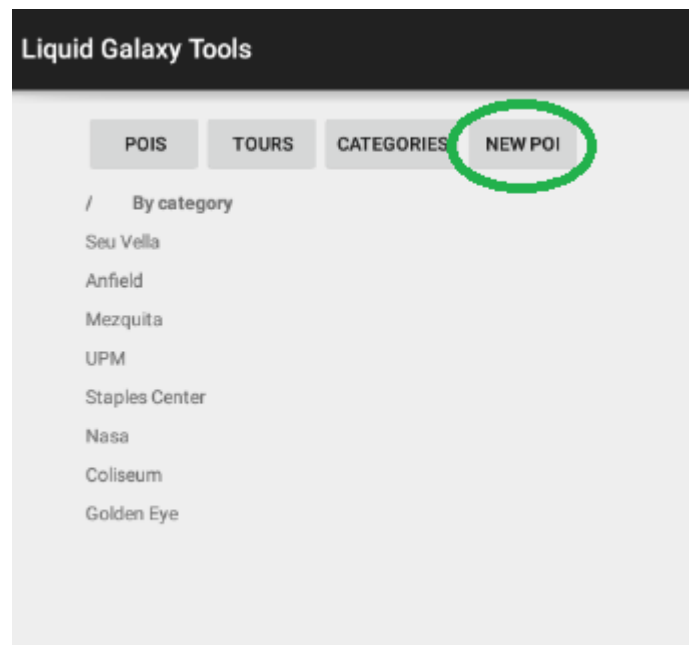


Figura 24: Botó per crear elements

Depèn de la secció que haguem seleccionat, el text del botó variarà:



Figura 25: Text per crear POI, Tour o Categoria

I si els ítems de la llista s'estan mostrant a partir de la vista per categories (filtrada), també apareix un botó per crear un nou ítem dins de la categoria on l'usuari es troba actualment:

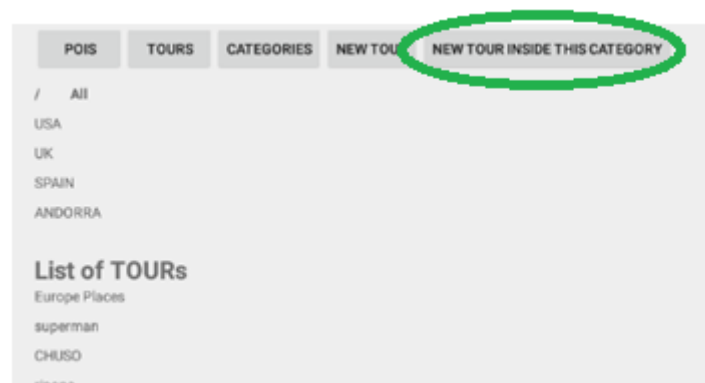


Figura 26: Nou element dins de la categoria actual

Quan es fa clic en un dels botons per crear un nou ítem s'obre una nova pantalla que varia segons el tipus seleccionat: POI, TOUR o CATEGORIA. L'explicació de què succeeix ve a continuació:

Nou POI

En aquesta pantalla apareixen tots els atributs necessaris que l'usuari ha d'omplir per a la creació d'un nou punt d'interès: nom de la ciutat, lloc visitat, altitud, longitud, latitud, heading, tilt, range, altitude mode i hide,

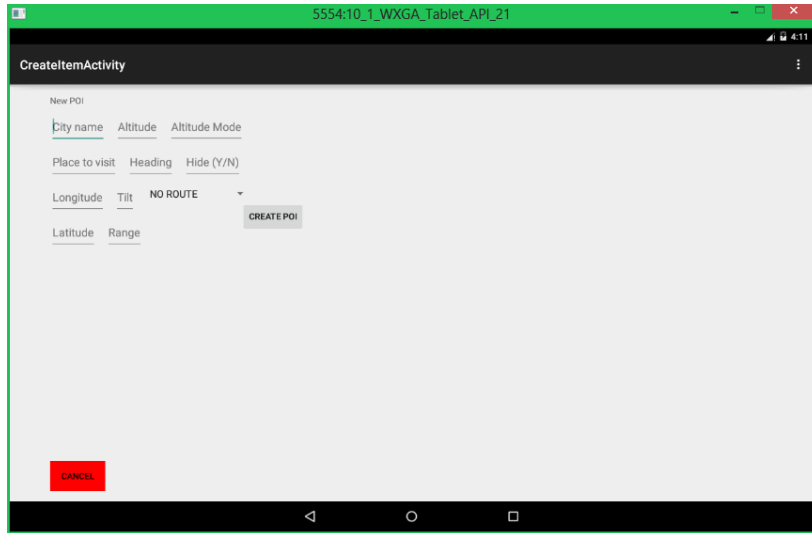


Figura 27: Nou POI

funció del qual consisteix en amagar la categoria a ulls de l'usuari. Per incloure el POI dins d'una categoria s'ha de seleccionar una del desplegable que hi ha, si no automàticament es guarda la selecció per defecte "NO ROUTE" i per tant aquest POI no queda associat a cap categoria. Si per la creació del POI haguéssim clicat l'opció "NEW POI INSIDE THIS CATEGORY" (crear POI en categoria actual), el desplegable per escollir la categoria no apareixeria ja que automàticament es guardaria la categoria on l'usuari es troba.

Com podem veure en la figura anterior, en la cantonada sud-oest hi ha un botó en vermell per a cancel·lar l'operació de creació.

Nova Categoria

La creació de categories és simple ja que l'usuari necessita aportar pocs camps: nom, valor per l'atribut *hide* i selecció de la categoria precedent.

Tal com passa amb els POIs i els Tours, si haguéssim escollit l'opció d'inserir una categoria dins de l'actual, en la pàgina de creació no apareixeria l'opció per seleccionar categoria i automàticament es guardaria la corresponent dins de la categoria on es trobi l'usuari navegant.

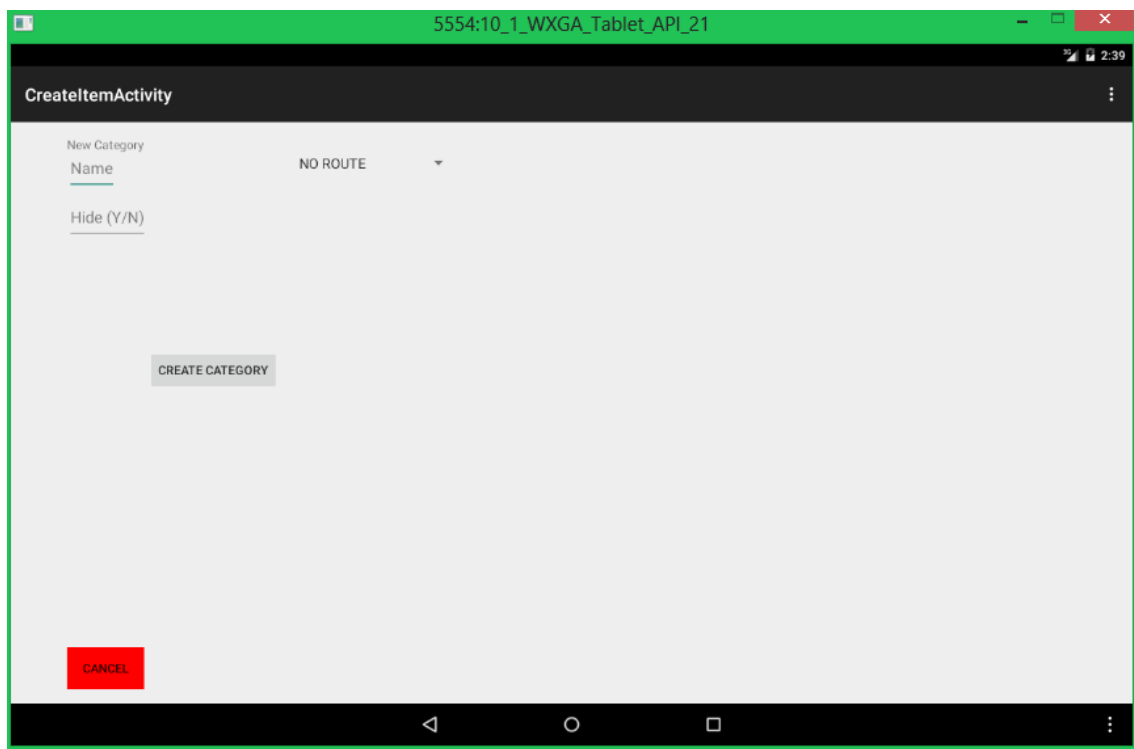


Figura 28: Nova Categoria

Nou Tour

Per afegir un nou Tour a la base de dades el procediment és una mica més complex.

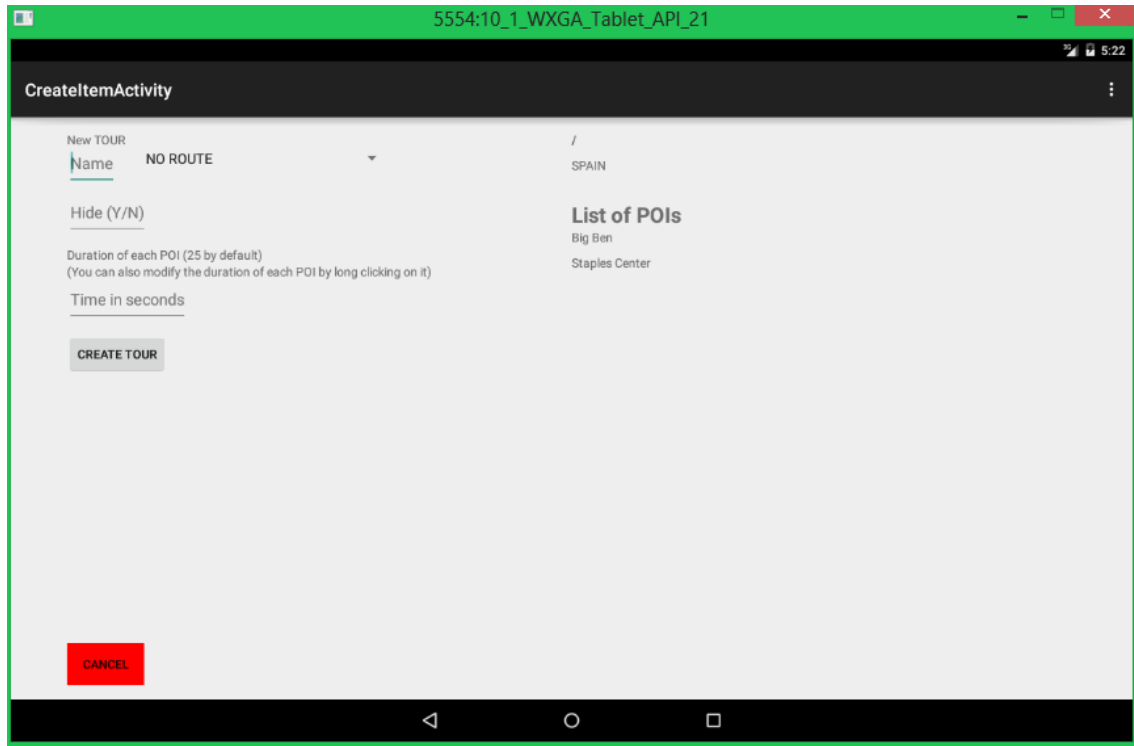


Figura 29: Nou Tour

La pantalla, com veiem en la figura, es divideix en dues parts: la de l'esquerra i la de la dreta.

A la part de l'esquerra l'usuari haurà d'omplir els camps corresponents al nom, valor corresponent a l'amagat, l'interval general de temps entre POI i POI, i seleccionar la categoria en què vol situar el Tour (si ha escollit l'opció d'inserir el Tour en la categoria on l'usuari es trobava, aquest últim camp no apareixerà). Però encara faltarà el més important d'un Tour: els punts d'interès.

A la part dreta apareixen tots els POIs guardats en l'aplicació tal com si un Usuari Bàsic estigués veient-los. En clicar un POI, apareix una petita finestra permetent a l'usuari afegir el POI en el Tour o cancel·lar l'operació.

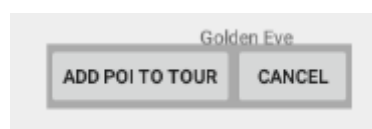


Figura 30: Afegir POI al Tour

Si l'usuari prem l'opció d'afegir un POI, aquest s'afegeix, tal com veiem en la figura següent, a una llista que es situa davall de la dels POIs persistents en la base de dades.

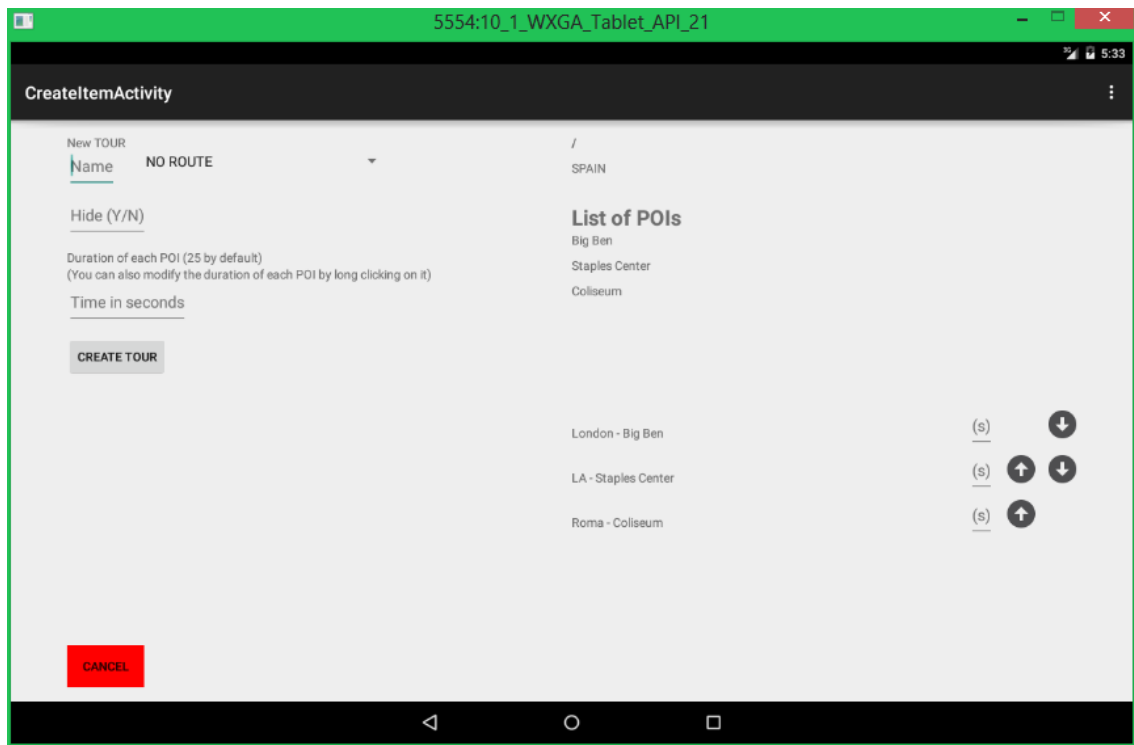


Figura 31: Llista de POIs en un Tour

Al costat de cada ítem d'aquesta llista, és a dir, de cada POI del Tour que l'usuari està creant, apareix un camp corresponent al temps de duració en segons (per això apareix escrit per defecte "(s)") que l'usuari vol aplicar a aquell POI en el Tour i unes icones de fletxes per canviar l'ordre dels POIs dins del Tour, una per baixar el POI de posició i una altra per pujar-lo.

Si l'usuari fa un clic llarg sobre un dels POIs de la llista del Tour apareix una nova finestra petita amb les opcions d'eliminar el punt de la llista o bé cancel·lar l'operació.

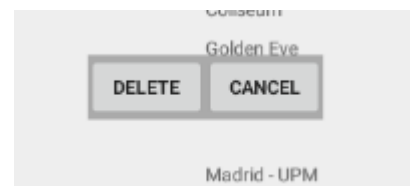


Figura 32: Eliminar POI del Tour

Modificació i Eliminació

Quan l'Usuari Administrador es troba en la pantalla d'administració i veu la llista de POIs, Tours o Categories, clicant en un ítem d'aquests pot modificar-lo (editar-lo).

Modificar o Eliminar POI

Si clics en un POI se li mostra una nova finestra petita amb les opcions d'Editar, Eliminar o Cancel·lar l'operació, tal com s'explica a l'inici de la secció "Usuari Administrador".

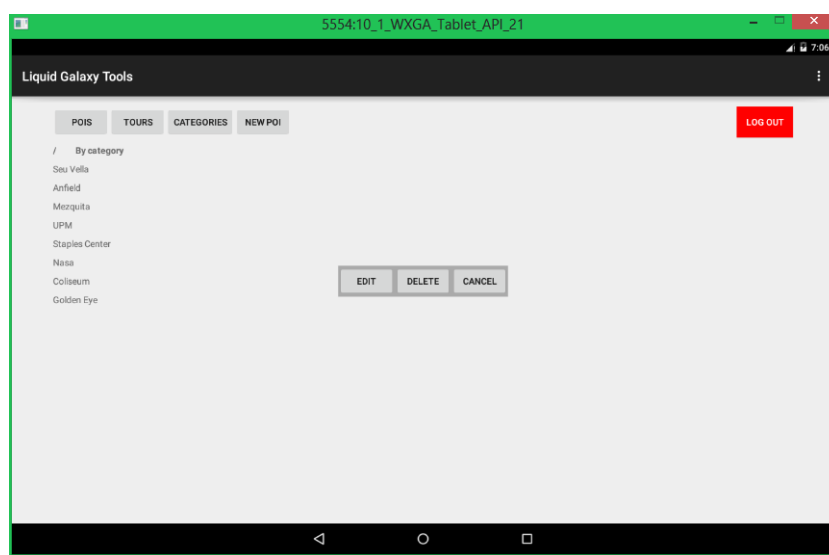


Figura 33: Modificar o Eliminar POI

Si l'usuari prem el botó d'eliminar (DELETE) el POI s'elimina de la base de dades de l'aplicació, per tant, deixa d'existir. Si l'elecció és la de modificar (EDIT), quan es prem el botó pertinent, s'obre una pantalla igual que la de creació de POIs, amb la diferència que els camps a omplir per l'usuari (els valors a modificar) tenen el valor per defecte corresponent al POI a editar.

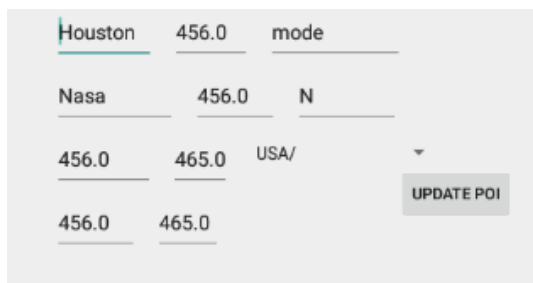
A screenshot of a form for modifying a POI. The form contains several input fields with pre-filled values: "Houston" (with a cursor), "456.0", "mode", "Nasa", "456.0", "N", "456.0", "465.0", "USA/", and "456.0", "465.0". There is a dropdown arrow next to "USA/" and an "UPDATE POI" button at the bottom right.

Figura 34: Modificar POI

Com es veu en la figura, tots els valors corresponen als atributs del POI que s'està editant i la categoria a la qual pertany és la que apareix per defecte en la llista de categories.

Modificar o Eliminar Tour

Igual que amb els POIs, en quan l'usuari clica en un Tour apareix la finestra per Editar, Eliminar o Cancel·lar. El botó d'Eliminar esborra el Tour, tal com passa amb els POIs, i el d'Editar també té el mateix comportament que amb els POIs, amb la diferència que en aquest cas obre la pantalla corresponent a modificació d'un Tour.

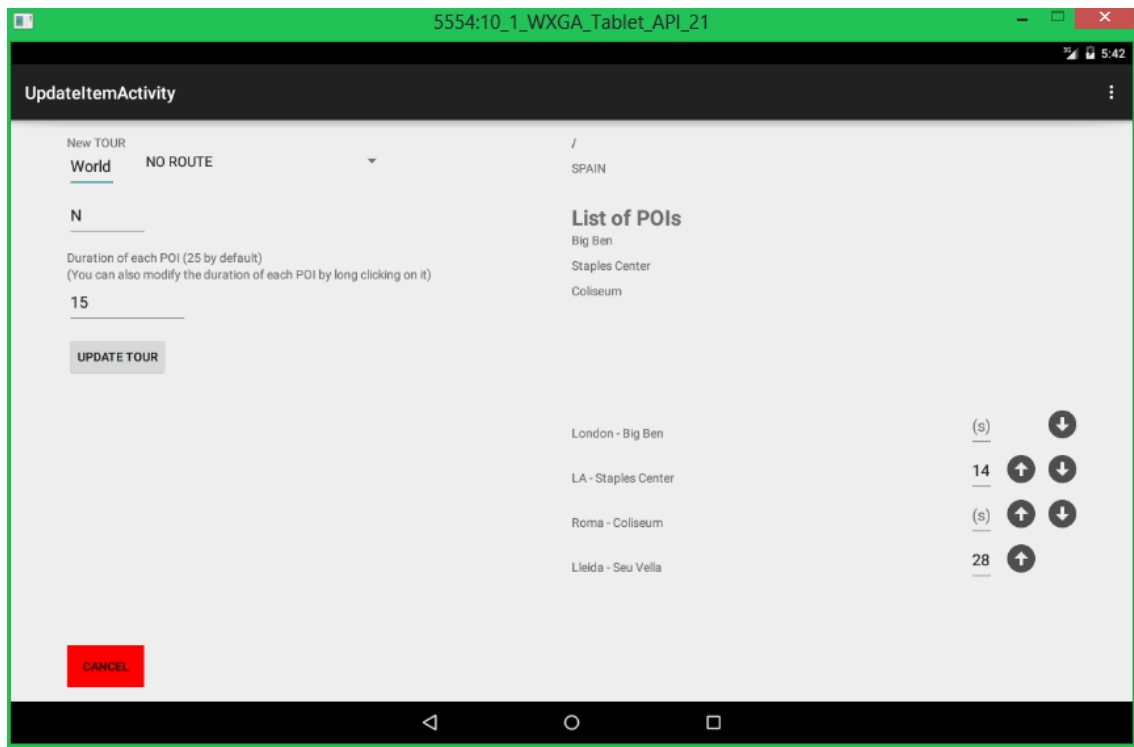


Figura 35: Modificar Tour

Tal com es pot veure en la figura, tots els atributs del Tour estan omplerts amb els valors corresponents al Tour que estem editant i també la llista de POIs que formen aquest. L'usuari pot modificar els camps escrits, canviar la categoria a la qual pertany i també afegir, eliminar o ordenar els punts d'interès pertanyents al Tour. També podem observar que alguns POIs tenen un temps de duració indicat per l'usuari i altres no. Aquests últims (tenen escrit per defecte "(s)") adquiriran el valor que l'usuari ha indicat com a interval general, situat en la part esquerra de la pantalla.

Modificar o Eliminar Categories

Per poder accedir a les opcions d'Editar i Eliminar passa el mateix que amb els POIs i Tours, però s'ha de tenir present quin tipus de vista tenim: si la vista és de totes les categories amb un simple clic ja podrem accedir a la finestra que ens dona les opcions desitjades; si la vista és filtrada per categories, amb un clic accedirem a les subcategories de l'ítem seleccionat però amb un clic llarg es mostraran les opcions desitjades.

L'usuari podrà canviar el nom de la categoria, el valor d'amagat i la categoria pare, és a dir, la categoria superior a la qual pertany. Si l'usuari decideix canviar la categoria pare, totes les subcategories de l'actual també s'arrossegaran i amb elles també els POIs i Tours de totes aquestes categories.

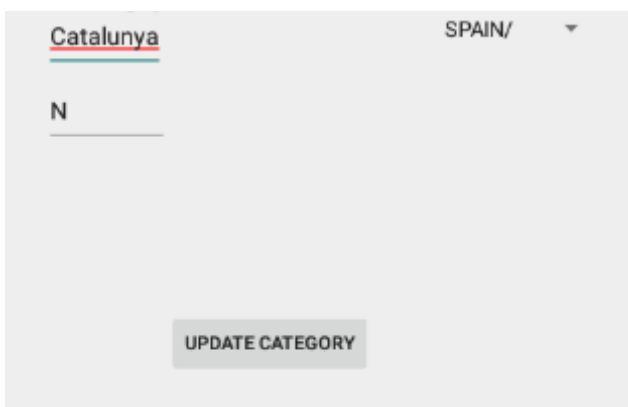


Figura 36: Modificar Categoria

A continuació expliquem un exemple per comprendre-ho millor en el que les categories “UdL”, “Campus Cappont” i “Facultat Medicina”, sigui amb més o menys profunditat, són subcategories de “Lleida” i al canviar la categoria pare de “UdL”, tal com hem explicat, també s'arrossegaran totes les seves subcategories i els seus POIs:

Categories

Nom Categoria	Categoria Pare	Shown Name
Lleida	-	Lleida/
UdL	Lleida	Lleida/UdL/
Campus Cappont	UdL	Lleida/UdL/Campus Cappont/
Facultat Medicina	UdL	Lleida/UdL/Facultat Medicina/
Tarragona	-	Tarragona/

Taula 5: Exemple Modificar Categoria - Taula Category

POIs

Nom POI	Categoria
Cafeteria	Lleida/UdL/Campus Cappelletti/
Cafeteria	Lleida/UdL/Facultat Medicina/
Servei d'Esports	Lleida/UdL/Campus Cappelletti/
Rectorat	Lleida/UdL/

Taula 6: Exemple Modificar Categoria - Taula POIs

La categoria pare de UdL és Lleida. Si l'usuari decideix canviar la categoria pare a Tarragona, tots els POIs d'aquesta i totes les subcategories i sub-POIs d'aquestes també es mouran de forma que quedaria així:

Categories

Nom Categoria	Categoria Pare	Shown Name
Lleida	-	Lleida/
UdL	Tarragona	Tarragona /UdL/
Campus Cappelletti	UdL	Tarragona /UdL/Campus Cappelletti/
Facultat Medicina	UdL	Tarragona /UdL/Facultat Medicina/
Tarragona	-	Tarragona/

Taula 7: Exemple Modificar Categoria - Taula Category Actualitzada

POIs

Nom POI	Categoria
Cafeteria	Tarragona/UdL/Campus Cappelletti/
Cafeteria	Tarragona/UdL/Facultat Medicina/
Servei d'Esports	Tarragona/UdL/Campus Cappelletti/
Rectorat	Tarragona/UdL/

Taula 8: Exemple Modificar Categoria - Taula POIs Actualitzada

Problemes i Solucions

Durant el desenvolupament del projecte ens hem trobat amb diverses dificultats, obstacles o decisions crucials que, d'una forma o altra, hem pogut anar resolent. Tots aquests poden classificar-se en tecnològics o d'implementació. Els classifiquem simplement perquè ens ajuda a identificar-los més ràpidament. Diversos dels problemes sorgits o decisions preses són:

Tecnològics

El primer problema va aparèixer mentre planificàvem el projecte i decidíem els objectius. Juntament amb l'Andreu, el tutor a l'empresa, teníem el dubte de fer l'aplicació per plataforma Android o per web. La que hi ha actualment, tot i tenir diversos errors, és via web però no s'acaba d'adaptar a totes les condicions necessàries establertes en els objectius del projecte. Per tant, vam decidir fer el canvi i realitzar-la per Android, així facilitaríem la seva implementació perquè el llenguatge de programació s'assembla molt a Java i durant el grau n'he practicat força. Un altre motiu es centrava en que la base de dades no portaria gaires embolics degut a que l'entorn Android Studio porta integrat funcionalitats de SQLite, i finalment també aconseguiríem que cada "corresponsal" d'un LG pogués descarregar-se l'aplicació gratuïta i crear els seus propis POIs.

Un altre problema ha estat relacionat amb la connexió de l'aplicació amb el Liquid Galaxy. Un cop vam decidir que seria via SSH, vam realitzar alguna prova de connexió però al laboratori on treballem (Lleida Liquid Galaxy LAB) vam tenir un problema de xarxa: el Wi-fi no funcionava. Per tant, vam haver de fer una prova a casa per sincronitzar dos ordinadors amb el Google Earth seguint uns passos indicats en una pàgina web [9].

Pel que fa referència a la implementació de la base de dades vam tenir que decidir entre *ADA Framework* o els ja mencionats *Content Providers*. L'ús de ADA hagués facilitat la crida d'algunes accions contra la base de dades però no gaudia de totes les funcionalitats aplicables que posseeixen els *Providers*, les quals he explicat en el apartat [Base de Dades](#). A més a més, al curs d'Udacity que vaig realitzar per aprendre Android utilitzaven aquests últims i això va ser un altre punt a favor per a l'ús dels proveïdors de contingut ja que dominava el seu ús.

D'implementació

Aquests problemes es presentaven a mesura que anàvem implementant les gestions d'administració dels POIs, dels Tours i de les Categories, així que detallarem alguns dels més costosos seguint aquest ordre.

POIs

Els problemes amb els punts d'interès sorgien en la forma que s'haurien de mostrar, en la forma que s'haurien de niar o classificar. Mostrar-los tots niats per categories implicava tenir totes les categories en una sola vista i això requeria molt espai. Per tant vam decidir classificar-los per categories i que solament es poguessin veure els POIs pertanyents a dita categoria. El problema era crear aquest conjunt de llistat de categories juntament amb el llistat de POIs i anar sincronitzant-los dinàmicament, és a dir, a mesura que l'usuari canviava de categoria. Això vam aconseguir-ho mitjançant l'ús d'un *layout*, que anomenaríem *fragment_pois.xml*, que es carregaria en pantalla cada cop que s'iniciés una instància de la classe *POISFragment.java*. Els elements clau d'aquest *layout* són dos objectes del tipus *ListView*, un per les categories i l'altre pels POIs.

<pre><ListView android:id="@+id/pois_listview" android:layout_width="wrap_content" android:layout_height="wrap_content" android:divider="@null" android:layout_below="@+id/pois_tittle_listview" android:visibility="gone"/></pre>	<pre><ListView android:id="@+id/categories_listview" android:layout_width="wrap_content" android:layout_height="wrap_content" android:scrollbars="vertical" android:divider="@null" android:layout_below="@+id/fragment_pois_route"/></pre>
--	---

Figura 37: Exemple codi ListViews

En cada un es carregaria la llista d'objectes corresponent i segons quina categoria clica l'usuari en la interfície de l'aplicació, es veuran uns elements o uns altres. Perquè aquesta càrrega de dades fos efectiva necessitava l'ús d'adaptadors. Els *adapters* són objectes que permeten carregar dades en diferents estructures de dades, en el nostre cas en estructures de tipus llista. Per tant vaig crear una classe anomenada *POIsAdapter.java* que estengués la classe *CursorAdapter*. Aquesta classe (l'adaptador) també s'utilitza per carregar els Tours en la seva llista corresponent, per tant el què fa és filtrar el tipus d'elements que s'han de carregar i mostrar-los a l'usuari en forma de llista.

El *layout* dels POIs també s'utilitza en la creació i modificació de Tours ja que l'usuari ha de poder veure de forma ordenada i classificada tots els POIs existents per poder afegir-los als Tours.

Tours

Aquests van donar força problemes pel que fa referència a l'agregació de POIs a l'itinerari d'un Tour. La part de la interfície d'usuari que s'encarregaria d'aquesta funcionalitat s'havia de dividir en dues parts: la corresponent a tots els POIs persistents en l'aplicació i la dels POIs del Tour en tractament. Per la llista de l'itinerari del Tour també hem utilitzat l'objecte de tipus *ListView* però amb més complexitat i hem creat un altre *adapter* anomenat *TourPOIsAdapter.java* el qual s'encarrega de tractar l'ordenació i la duració dels POIs d'un Tour. El problema succeïa quan l'usuari volia reordenar els punts d'interès. Vam indagar per la xarxa i vam trobar una vídeo explicatiu [10] de com moure els elements d'una llista arrossegant-los però, un cop aplicava l'algoritme en el nostre projecte, per raons desconegudes fallava. Vam intentar resoldre'l modificant l'algoritme que em proporcionaven i modificant el propi codi de l'aplicació però l'error persistia. Així que, per manca de temps, vam decidir canviar de mètode de forma que cada ítem de la llista disposaria de dos botons en forma de fletxa per ordenar cada POI, tal com s'explica en la secció Usuari Administrador, apartat de [Creació](#).

Però els problemes no acabaven aquí. Un cop l'usuari ordena els POIs, l'aplicació ha de saber que l'ordre d'aquests ha variat. Com que la classe encarregada d'enregistrar els POIs en la base de dades no tenia opció de recuperar la llista ordenada, vam haver de fer ús del mètode *notifyChanges()* que facilita la classe *BaseAdapter*, la qual estén el nostre adaptador *TourPOIsAdapter*.

Categories

Quan l'usuari movia una categoria, és a dir, quan canviava la categoria a la qual en pertanyia una altra, totes les subcategories de la moguda també s'haurien de moure i això implicava un cert embolic. Les dificultats apareixien en la modificació del camp *Shown_Name*, significat del qual he explicat en la secció "Base de Dades", LINK apartat "Category". Per tant, l'aplicació havia de mirar si existien subcategories i moure-les. Posteriorment, mirar si existien subcategories d'aquestes subcategories anteriors, tornar a moure-les i així successivament amb totes les categories pertanyents. Realitzar aquesta operació era un mètode costós i vam tenir molts problemes perquè aquesta operació finalitzés correctament. Gràcies a la recursivitat vam poder aconseguir que totes es moguessin sense cap problema però era força costós així que vam crear una consulta SQL que directament m'obtingués totes les categories que continguessin la ruta a modificar. El codi encarregat d'aquesta operació és:

```
private void updateSubCategoriesShownName(ViewHolderCategory viewHolderCategory, String oldItemShownName){

    String whereClause = POIsContract.CategoryEntry.COLUMN_SHOWN_NAME + " LIKE '" + oldItemShownName + "%'";

    Cursor cursor = getActivity().getContentResolver().query(POIsContract.CategoryEntry.CONTENT_URI,
        new String[]{POIsContract.CategoryEntry._ID, POIsContract.CategoryEntry.COLUMN_SHOWN_NAME},
        whereClause, null, null);

    ContentValues updatedShownNameValues;
    String currentShownName, finalShownName, itemTreatedID;
    int updatedRows = 0;
    while(cursor.moveToNext()){

        itemTreatedID = String.valueOf(cursor.getInt(0));
        currentShownName = cursor.getString(1);
        if(!currentShownName.equals(oldItemShownName)) {
            //remove the bad shownName
            String currentWithoutOldPartition = currentShownName.substring(oldItemShownName.length(), currentShownName.length());
            //write the good one
            finalShownName = newShownName + currentWithoutOldPartition;

            updatedShownNameValues = new ContentValues();
            updatedShownNameValues.put(POIsContract.CategoryEntry.COLUMN_SHOWN_NAME, finalShownName);
            updatedRows += getActivity().getContentResolver().update(POIsContract.CategoryEntry.CONTENT_URI, updatedShownNameValues,
                Category_IDselection, new String[]{itemTreatedID});
        }
    }

    Toast.makeText(getActivity(), String.valueOf(updatedRows), Toast.LENGTH_SHORT).show();
    Intent adminIntent = new Intent(getActivity(), AdminActivity.class);
    startActivity(adminIntent);
}
```

Figura 38: Exemple codi modificar subcategories

Extensions al TFG en el marc de GSoC

Tot i que, com hem mencionat en l'apartat anterior, el més complex ja està realitzat, encara queden una sèrie d'aspectes a desenvolupar:

- Una atractiva i moderna interfície d'usuari seguint els estàndards del *Material Design* de Google.
- Implementar la funcionalitat de cerca de POIs manual.
- Un cercador de punts d'interès per veu.
- Inserir un mapa del Google Maps perquè es puguin cercar i seleccionar POIs en aquest i es mostrin en els monitors del Liquid Galaxy.
- Afegir una funcionalitat per importar POIs des d'un fitxer extern a l'aplicació.
- Afegir les opcions per executar funcionalitats internes de gestió del Liquid Galaxy.
- Millorar i optimitzar el codi.
- Crear un manual de funcionament en línia.

Avaluació, valoració i conclusions

La part més complexa de l'aplicació, que és tota la gestió dels POIs, Tours i Categories per part de l'Usuari Administrador, ja està feta. Estem molt contents de veure que els resultats han sortit com havíem pensat i preparat anteriorment, és a dir, els objectius inicials s'estan complint satisfactòriament. Alguns aspectes inclús millor del que esperàvem.

Personalment, penso que l'esforç dedicat cada matí, tarda i algunes nits de pràcticament cada dia de la setmana durant els últims mesos han servit per realitzar aquesta part de rerefons de l'aplicació que l'usuari no veu però que és bàsica pel seu eficient funcionament. També crec que aquests resultats són fruit, en part, de la meua implicació i del fet que estic gaudint molt amb aquest projecte: implementant l'aplicació, aprenent llenguatges i coneixements nous, descobrint noves formes de treballar tant individuals com col·lectives, interactuant amb altres membres i participants del GSoC, etc. El desenvolupament d'aquest TFG no se m'ha fet gens pesat i això ha ajudat molt a la seva, segons el meu punt de vista, bona execució.

Estic content d'haver solucionat satisfactòriament tots els problemes sorgits, alguns amb solucions òptimes i altres gràcies a l'ús de diferents recursos, força dels quals han estat elaboracions meves i altres resultat d'indagar per internet o consultar a altres companys.

Com hem mencionat en l'apartat anterior, encara hi ha aspectes a realitzar i millorar, tasques que posteriorment a la presentació del projecte ja començarem a realitzar i d'altres que s'hauran d'optimitzar perquè finalment pugui obtenir un producte realment eficient i utilitzable.

Per finalitzar, vull tornar a dir que em sento orgullós del projecte realitzat i amb ganes d'evolucionar-lo fins al seu estat òptim, perquè realment he estat i estic gaudint molt amb el desenvolupament del Liquid Galaxy POIs Controller.

Bibliografia

- [1] R. D. Sàez, «GitHub,» 2015. [En línia]. Available: https://github.com/rafads7/Liquid_Galaxy_POIs_Controller.
- [2] G. Melange, «Google Summer of Code 2015,» 2015. [En línia]. Available: <https://www.google-melange.com/gsoc/homepage/google/gsoc2015>.
- [3] Google, «Google Liquid Galaxy,» [En línia]. Available: <http://www.google.com/earth/explore/showcase/liquidgalaxy.html>.
- [4] IEEE, «Normas IEEE,» 1884. [En línia]. Available: <http://normasieee.com/>. [Últim accés: 2015].
- [5] A. Ibáñez, «Lleida Liquid Galaxy LAB,» [En línia]. Available: <http://www.liquidgalaxylab.com/>. [Último acceso: 2015].
- [6] Udacity, «Developing Android Apps,» 2015. [En línia]. Available: <https://www.udacity.com/course/developing-android-apps--ud853>.
- [7] M. X, «Miriada X,» Universidad Politécnica de Valencia, 2013. [En línia]. Available: https://www.miriadax.net/web/android_programacion. [Últim accés: 2015].
- [8] R. D. Sàez, «GitHub, Sunshine,» 2015. [En línia]. Available: <https://github.com/rafads7/Sunshine>.
- [9] G. Code, «Liquid Galaxy Quick Start,» [En línia]. Available: <https://code.google.com/p/liquid-galaxy/wiki/QuickStart>. [Último acceso: 2015].
- [10] A. Developers, «ListView Cell Dragging,» 2013. [En línia]. Available: https://www.youtube.com/watch?v=_BZlvjMgH-Q. [Último acceso: 2015].
- [11] U. d. Lleida, «Universitat de Lleida,» [En línia]. Available: <http://www.udl.cat/>.
- [12] P. 2002, «Ponent 2002,» [En línia]. Available: <http://www.ponent2002.com/>.